CRANFIELD UNIVERSITY

Michael Farnsworth

Multi-level and Multidisciplinary Optimisation of Microelectromechanical Systems

Manufacturing and Materials Department

School of Applied Sciences

PhD

Academic Year: 2008 – 2011

Supervisors: Prof. Ashutosh Tiwari and Dr Meiling Zhu

CRANFIELD UNIVERSITY

Manufacturing and Materials Department

School of Applied Sciences

PhD Thesis

Academic Year: 2008 – 2011

Michael Farnsworth

Multi-level and Multidisciplinary Optimisation of Microelectromechanical Systems

Supervisors: Prof. Ashutosh Tiwari and Dr. Meiling Zhu

October 2012

This thesis is submitted in fulfilment of the requirements for the degree of Doctor of Philosophy.

Abstract

A comparative investigation into the role multi-level and multidisciplinary design optimisation can play in the automated design synthesis of microelectromechanical systems (MEMS) is presented. Microelectromechanical systems are a field grown out of the integrated circuit industry, with the goal of developing smart micro devices which can interact with the environment in some form. They promise to revolutionise our present day lifestyles as much as the integrated circuit has done in recent decades.

The complexity in fabrication, the delicacy in size that each device encompasses and the multidisciplinary nature means design synthesis is a highly complicated process. Current challenges stemming from their design include the high levels of computational cost required in their modeling and analysis, and the often increasing complexity of design through the coupling of multiple components and devices into a functioning system.

The development of automated design synthesis tools and methodologies to aid MEMS design is therefore important to overcome these challenges in order to accommodate the growing field of MEMS as it expands into more and more areas and continues opening up to more and more applications.

An update of the current state of the art in automated MEMS design synthesis and optimisation is first presented, utilizing state of the art multi-objective evolutionary algorithms over five separate MEMS design optimisation case studies. The field of multi-level and multidisciplinary optimisation is critically reviewed and discussed with respect to their application to MEMS design synthesis and optimisation. The outcome is twofold, with the construction of both a novel multidisciplinary optimisation algorithm tailored towards MEMS design and a set of multi-level design optimisation strategies.

This thesis next outlines and develops a novel modular soft computing framework to house the multi-objective, multi-level and multidisciplinary design optimisation strategies. In order to evaluate both the current state of the art in automated MEMS design synthesis and the multi-level and multidisciplinary optimisation strategies outlined a hierarchical MEMS bandpass filter case study has been constructed. Incorporating a novel state of the art electrical equivalent modelling and design synthesis approach, six novel design problems structured around the MEMS bandpass filter were developed and formed the basis for the comparative study to follow.

Finally both the current state of the art in automated MEMS design synthesis, multi-objective evolutionary algorithms, and the outlined and developed multi-level and multidisciplinary optimisation strategies are applied to the six design problems developed. Comparative analysis and discussion is then given, showing a marked improvement in MEMS design synthesis for the multi-level and multidisciplinary optimisation strategies over the current state of the art methodology.

Acknowledgments

First of all, I would like to thank my advisors, Professor Ashutosh Tiwari and Dr. Meiling Zhu, for their kind support, encouragement and guidance. Professor Tiwari in particular deserves my warmest thanks for his support in the development of this research and thesis, and the many opportunities for professional development.

I am also deeply indebted to a kind friend Dr. Elhadj Benkhelifa who was also with me along this particular journey. His enthusiasm, constructive criticism and fruitful discussions with out which this thesis would be sorely diminished are most kindly acknowledged.

Finishing a Ph.D can be a lonely task at times and requires more than simply doing research. I am very grateful to the open discussion and dialogue found in Professor Ramsden's research group, in particular to my good friend Dr. Gergely Bandi. Gergely and I started this voyage of discovery at the same time and he has been an incredibly supportive and enlightening friend, who also I should note finished before me, which is one of many bets I have lost.

Finally I must give special thanks to my supportive family, my mother for her encouragement and love, and my brothers and sister for their continued support and generosity. A special thank you to Alex my twin brother and equal, even if you did manage to submit before me. I couldn't have done this without you.

Never trust a man wearing a better suit than your own

- 47th Rule of Acquisition

Table of Contents

List of Publications

Journal Publications

- M., Farnsworth, E., Benkhelifa, A. Tiwari, M., Zhu, and M. Moniri, "*An Efficient Evolutionary Multi-Objective Framework for MEMS Design Optimisation: Validation, Comparison and Analysis*" Journal of Memetic Computing, 2011, Vol 3 pp. 175-197
- E., Benkhelifa, M., Farnsworth, A., Tiwari, and M., Zhu, "*Design and optimisation of microelectromechanical systems: a review of the state-of-the-art* " International Journal of Design Engineering 2010 - Vol. 3, No.1 pp. 41-76
- M., Farnsworth, E., Benkhelifa, A. Tiwari, and M., Zhu, "*A Novel Approach to Multi-level Evolutionary Design Optimization of a MEMS Device*" Lecture Notes in Computer Science, 2010, Volume 6274/2010, 322-334, DOI: 10.1007/978-3-642-15323-5_28
- M., Farnsworth, E., Benkhelifa, A. Tiwari, and M., Zhu, "*A Multi-level Evolutionary Design Optimisation of MEMS: Methodology and Application*" International Journal of Simulation and Multidisciplinary Design Optimisation. Accepted to be published.

Book Chapter Publications

- E., Benkhelifa, M., Farnsworth, A. Tiwari, and M., Zhu, "*Evolutionary Algorithms for Planar MEMS Design Optimisation: A Comparative Study*" Studies in Computational Intelligence, 2010, Volume 284/2010, pp.199-210, DOI: 10.1007/978-3-642-12538-6_17

Conference Publications

- M., Farnsworth, E., Benkhelifa, A. Tiwari, and M., Zhu, "*A Novel Approach to Multi-level Evolutionary Design Optimization of a MEMS Device*" International Conference From Biology to Hardware, Evolvable Systems ICES 2010, York, UK.
- M., Farnsworth, E., Benkhelifa, A. Tiwari, and M., Zhu, "*A Multi-level Evolutionary Design Optimisation of MEMS: Methodology and Application*" In Proceedings Third International Conference on Multidisciplinary Design Optimization and Applications 2010
- E., Benkhelifa, A., Tiwari, and M., Farnsworth "*Modefrontier a Facilitator for MEMS Design Optimisation Integration*" International Modefrontier User Meeting, 17-28 May 2010, Trieste, Italy
- E., Benkhelifa, M., Farnsworth, A. Tiwari, and M., Zhu, "*Evolutionary Algorithms for Planar MEMS Design Optimisation: A Comparative Study*" International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO 2010) 12-14 May 2010, Granada, Spain
- E., Benkhelifa, M., Farnsworth, A., Tiwari, and M., Zhu, *"An Integrated Framework for MEMS Design Optimisation using modeFrontier*" EnginSoft International Conference 2009, CAE Technologies For Industry and ANSYS Italian Conference 2009.

List of Figures

List of Tables

List of Equations

$$K_x = \frac{2Et(W_b)^3}{{L_b}^3} \frac{{L_t}^3 + 14\alpha L_t L_b + 36\alpha^2 {L_b}^2}{4{L_t}^2 + 41\alpha L_t L_b + 36\alpha^2 {L_b}^2} \quad \text{(A.1)}$$

$$\alpha = \left(\frac{W_t}{W_b}\right)^3 \quad \text{(A.2)}$$

$$B_x = \mu\left[\left(A_s + 0.5A_t + 0.5A_b\left(\frac{1}{d} + \frac{1}{\delta}\right) + \frac{A_c}{g}\right)\right] \quad \text{(A.3)}$$

$$m_x = m_s + \frac{1}{4}m_t + \frac{12}{35}m_b \quad \text{(A.4)}$$

$$where\ m_s = \rho A_s t \qquad m_t = \rho A_t t \qquad m_b = \rho A_b t \quad \text{(A.5)}$$

$$S = \frac{2mF_e\omega}{K_z K_x} \frac{1}{\sqrt{(1-{r_z}^2)^2 + \left(\frac{r_z}{Q_z}\right)^2}\sqrt{(1-{r_x}^2)^2 + \left(\frac{r_x}{Q_x}\right)^2}} \quad \text{(A.6)}$$

Electrostatic force:
$$F_e = 2\frac{2(N+1)\varepsilon_0 h}{g} V_{ac}V_{dc} \quad \text{(A.7)}$$

Driving mode natural frequency:
$$\omega_{rx} = \sqrt{\frac{K_x}{m}} \quad \text{(A.8)}$$

Sensing mode natural frequency:
$$\omega_{rz} = \sqrt{\frac{K_z}{m}} \quad \text{(A.9)}$$

Driving mode frequency ratio:
$$rx = \frac{\omega}{\omega_{rx}} \quad \text{(A.10)}$$

Sensing mode frequency ratio:
$$rx = \frac{\omega}{\omega_{rx}} \quad \text{(A.11)}$$

$$0 \le L_{cy} + 2g + 2W_c \le 700\mu m \quad \text{(A.12)}$$

$$0 \le L_{cy} + 2L_b + 2W_t \le 700\mu m \quad \text{(A.13)}$$

$$0 \le 3L_t + W_{sy} + 4L_C - 2X_0 + 2W_{cy} + 2W_{ca} \le 700\mu m \quad \text{(A.14)}$$

$$(2N+1)W_c + 2N_g \le L_{cy} \quad \text{(A.15)}$$

$$4 \le L_c - \left(X_0 + X_{disp}\right) \le 200\mu m \quad \text{(A.16)}$$

Where:

$$X_{disp} = \frac{QF_{e,x}}{Kx} \quad \text{(A.17)}$$

$$F_{e,x} = 1.12\varepsilon_0 N\frac{1}{g}V^2 \quad \text{(A.18)}$$

$$Q_x \geq 5 \tag{A.19}$$

$$Q_z \geq 5 \tag{A.20}$$

$$V_{dc} + V_{ac} < V_{si} \tag{A.21}$$

$$Z_{max} \leq 0.1\mu m \tag{A.22}$$

$$X_{max} \leq 5\mu m \tag{A.23}$$

Where:

$$K_y = \frac{4EhB_w}{B_l} \tag{A.24}$$

$$V_{si} = \frac{g^2 K_X}{2N\varepsilon_0 W_{finger}}\left(\sqrt{2\frac{K_y}{K_x} + \frac{{X_0}^2}{g^2}} - \frac{X_0}{g}\right) \tag{A.25}$$

$$\begin{gathered} f_1 = 0.7854\, x_1 {x_2}^2 \left(\frac{10{x_3}^2}{3} + 14.933\, x_3 - 43.0934\right) - 1.508\, x_1({x_6}^2 + {x_7}^2) \\ + 7.477({x_6}^3 + {x_7}^3) + 0.7854(x_4 {x_6}^2 + x_5 {x_7}^2) \\ f_2 = \frac{\sqrt{\left(\frac{745\, x_4}{x_2 x_3}\right)^2 + 1.69 \times 10^7}}{0.1{x_6}^3} \qquad f_3 = \frac{\sqrt{\left(\frac{745\, x_5}{x_2 x_3}\right)^2 + 1.575 \times 10^7}}{0.1{x_7}^3} \end{gathered} \tag{4.1}$$

$$\begin{gathered} g_1 \equiv \frac{1}{x_1 {x_2}^2 x_3} - \frac{1}{27} \leq 0; \quad g_2 \equiv \frac{1}{x_1 {x_2}^2 {x_3}^2} - \frac{1}{397.5} \leq 0; \\ g_3 \equiv \frac{{x_4}^3}{x_2 x_3 {x_6}^4} - \frac{1}{1.93} \leq 0; \quad g_4 \equiv \frac{{x_5}^3}{x_2 x_3 {x_7}^4} - \frac{1}{1.93} \leq 0; \\ g_5 \equiv x_2 x_3 - 40 \leq 0; \quad g_6 \equiv \frac{x_1}{x_2} - 12 \leq 0; \\ g_7 \equiv 5 - \frac{x_1}{x_2} \leq 0; \quad g_8 \equiv 1.9 - x_4 + 1.5x_6 \leq 0; \\ g_9 \equiv 1.9 - x_5 + 1.1x_7; \quad g_{10} \equiv f_2 - 1800 \leq 0; \\ g_{11} \equiv \frac{\sqrt{\left(\frac{745\, x_5}{x_2 x_3}\right)^2 + 1.575 \times 10^7}}{0.1{x_7}^3} - 1100 \leq 0; \end{gathered} \tag{4.2}$$

$$i_n = -V_{Pn}\frac{\partial C_n}{\partial t} = -V_{Pn}\frac{\partial C_n}{\partial x}\frac{\partial x}{\partial t} \tag{5.1}$$

$$\frac{\partial C_n}{\partial x} = \frac{2\xi N_{fin}\varepsilon_o h}{d} \tag{5.2}$$

$$F = m\frac{du}{dt} = m\frac{dx^2}{dt^2} \tag{5.3}$$

$$v = L\frac{di}{dt} = L\frac{dq^2}{dt^2} \tag{5.4}$$

$$R_{xn} = \frac{c_{rs}}{\eta_{en}^2} = \frac{\sqrt{k_{rs}m_{rs}}}{Q\eta_{en}^2} \tag{5.5}$$

$$L_{xn} = \frac{m_{rs}}{\eta_{en}^2} \tag{5.6}$$

$$C_{xn} = \frac{\eta_{en}^2}{k_{rs}} \tag{5.7}$$

$$\eta_{en} = V_{pn}\frac{\vartheta C_n}{\vartheta x} \tag{5.8}$$

$$\frac{\vartheta C_n}{\vartheta x} = \frac{2\xi N_{fin}\varepsilon_o h}{d} \tag{5.9}$$

Mechanical error:

$$\left[\left(\frac{1}{2.45}\right)\cdot\ |(Nodal\ Stiffness\ Kx\ Error)|\right] + \left[\left(\frac{1}{5.12e^{-10}}\right)\cdot\ |(Mass\ Error)|\right] + \left[\left(\frac{1}{21.55}\right)\cdot\ |(Comsol\ Stiffness\ Kx\ Error)|\right] \tag{7.1}$$

List of Acronyms

AC	Alternating Current
API	Application Programming Interface
BEM	Boundary Element Method
BEA	Boundary Element Analysis
CAD	Computer-Aided-Design
CAE	Computer-Aided-Engineering
CBR	Case Based Reasoning
CMOS	Complementary Metal-Oxide-Semiconductor
DC	Direct Current
DAG	Direct Acyclical Graph
DGA	Distributed Genetic Algorithm
EA	Evolutionary Algorithm
EC	Evolutionary Computation
FDM	Finite Difference Method
FEA	Finite Element Analysis
GA	Genetic Algorithm
GAECM	Genetic Algorithm Electric Circuit Model
GP	Genetic Programming
GPBG	Genetic Programming Bond Graph
HDL	Hardware Description Language
HF	High Frequency
IC	Integrated Circuit
IF	Intermediate Frequency
IHC	Interactive Hybrid Computation
IMEA	Island Model Evolutionary Algorithm
IPE	In-exact Pre Evaluation
MDO	Multidisciplinary Optimisation
MA	Memetic Algorithm
MEMS	Micro-Electro-Mechanical System
MOEA	Multi-Objective Evolutionary Algorithm
MOGA	Multi-Objective Genetic Algorithm
MOR	Model-Order Reduction
NODAS	Nodal Design of Actuators and Sensors
NSGAII	Non Sorting Genetic Algorithm II
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
Q	Quality
RF	Radio Frequency
SAW	Surface Acoustic Wave
SBX	Simulated Binary Crossover
SPEA2	Strength Pareto Evolutionary Algorithm 2
SQP	Sequential Quadratic Programming
VLSI	Very-Large-Scale-integration

1

Introduction

This thesis considers the role multi-level and multidisciplinary optimisation can play in the automated design synthesis of microelectromechanical systems (MEMS). This chapter introduces background information on the field MEMS and presents gaps within the process of MEMS design and motivations to solve these gaps. Following this are a number of research thrusts that outline the main aims and objectives for investigating and overcoming gaps within the field of MEMS design. The remainder of the chapter presents some of the constraints placed upon this research, the contributions born out of this work, and an outlined of the thesis as a whole.

1.1 Background

Micro-electro-mechanical systems (MEMS) or micro-machines [1] are a field grown out of the integrated circuit (IC) industry, utilizing fabrication techniques from the technology of Very-Large-Scale-Integration (VLSI) to in essence add micromechanical elements to electrical circuits. The goal is to incorporate the physical world of perception and control into electrical systems to develop smart devices which can interact with their environment in some form.

The paradigm of MEMS is well established within both the commercial and academic fields; at present encompassing more than just the mechanical and electrical [2], MEMS devices now cover a broad range of disciplinary domains, including the fluidic, thermal, chemical, biological and magnetic systems. This has resulted in a host of applications to arise, from micro-resonators [3], gyroscopes [4], micro-fluidic devices [5][6], biological lab on chip devices [7][8], pressure sensors [9][10], optical switches [11][12] and micro-relays [13] to name but a few.

The devices developed have inherent advantages over their macro counterparts with smaller, lighter and often more responsive components, their cost is greatly reduced as a result and their survivability and lifespan are often much greater than the bulkier macro equivalents[1]. This is in part due to packaging of the device which can protect it from the elements, the reduction in moving mechanical components required for function and the strength of material used in relation to the physical forces played out at a MEMS scale. Silicon-based films, for example polycrystalline-silicon, are the dominant structural material for MEMS devices, giving ability for integrated mechanical and electronic components.

It is no surprise that research and development in MEMS is growing year on year and is of crucial importance to the world economy as an emerging market for innovation and design.

The conventional approach to MEMS design is often a non-automated hand-driven methodology where the creation of a model in some chosen representation is simulated based upon the physical behavior associated with it and later is visualized and analyzed.

In the recent years there has been an interest in standardizing the design practices in the MEMS domain, however currently design synthesis methodologies for MEMS are undefined with regards to the design of an unambiguous target [14]. Though standard models do exist they generally provide only an outline for a design strategy, an illustration of which is given by Fedder [15]. Here an outline of an overall system specification is defined and built upon the functional requirements of the device as specified by the designer. This is used as a guide for the construction of a higher level system description designed to match the functional requirement behavior and consisting of abstract models for all but the simplest components that make up the device. The next step then often looks at developing the micromechanical 2D layout through to the solid modeling of the device or components before eventually leading to fabrication and testing.

Fabrication of the micromechanical components of the device often fall upon 'micromachining' processes utilizing mask layouts of the design to selectively etch away designated parts of the silicon wafer or add new structural layers to form the mechanical and/or electromechanical devices [16].

Traditional development of MEMS by silicon micromachining fabrication techniques [17] requires both many prototypes and a long line of experimentation (design process). This often involves similar methodologies to those described above with mask layout designs produced in a trial and error approach dependant on user experience and through their fabrication, testing and subsequent redesign a device can be built. This approach nominally coined a 'Build and Break' iterative is both time-consuming and expensive.

A number of Computer-Aided-Design / Engineering (CAD/E) tools and simulators have been developed and used to facilitate an improvement in the design process [18][19][20]. However this does not solve a fundamental problem with the current approach to MEMS design synthesis, the dependence on hand-driven design and optimisation. The manual design of a MEMS device can be undertaken by an experienced designer however due to the multifaceted nature of a MEMS device, experience in one field of design may not be sufficient as it may encompass several different domains or disciplines.

The development of automated design synthesis tools and methodologies to aid MEMS design is important in order to accommodate the growing field of MEMS as it expands into more and more areas and continues opening up to more and more applications.

The complexity in fabrication, the delicacy in size that each device encompasses means design synthesis is a highly complicated domain, which is generally broken up between many highly skilled professionals and where available support tools. Therefore the need for computer aided

design (CAD) tools to assist, speed up and optimize the design synthesis process is of great importance.

1.2 Problem statement

In 1996 a group of leading academics and industrial experts convened on a workshop sponsored by the National Science Foundation in the USA looking into 'structured design methods for MEMS'. The primary question posed to the workshop was the following: "Can successes in developing structured design methods for digital VLSI be extended into the domain of MEMS? If so, what lessons could be learnt and transferred" [21].

The scale and growth of the microelectronics industry has been huge over the past 40 years providing an enormous impact on the daily lives of normal people. This 'VLSI' revolution has been helped and driven by the advancement in the semiconductor process technology, the creation of automated design tools and simulation models and a structured hierarchical design methodology [21].

Key to this rapid success was the early definition of a clean digital interface that separated design efforts at increasingly high levels of abstraction from the growing complexities of the fabrication process [21]. However though there have been recent advances in MEMS manufacturing, the need for a structured design methodology remains elusive [14].

Several elements which contributed to the development and early success of VLSI structured design sadly do not transcend into the field of MEMS. However three areas common in both VLSI design and MEMS design were identified as, 'Languages', 'Libraries', and 'Simulation', each of which was important in the development of design methodologies for VLSI and in MEMS could provide the foundation for (semi-)automatic synthesis of MEMS devices [21].

An early example of a possible design and analyses process was outlined in the final report and is shown in figure 1.1, showing briefly how a mask layout of a device can be simulated to create a shape which can then also be simulated and analysed for its functional behaviour.

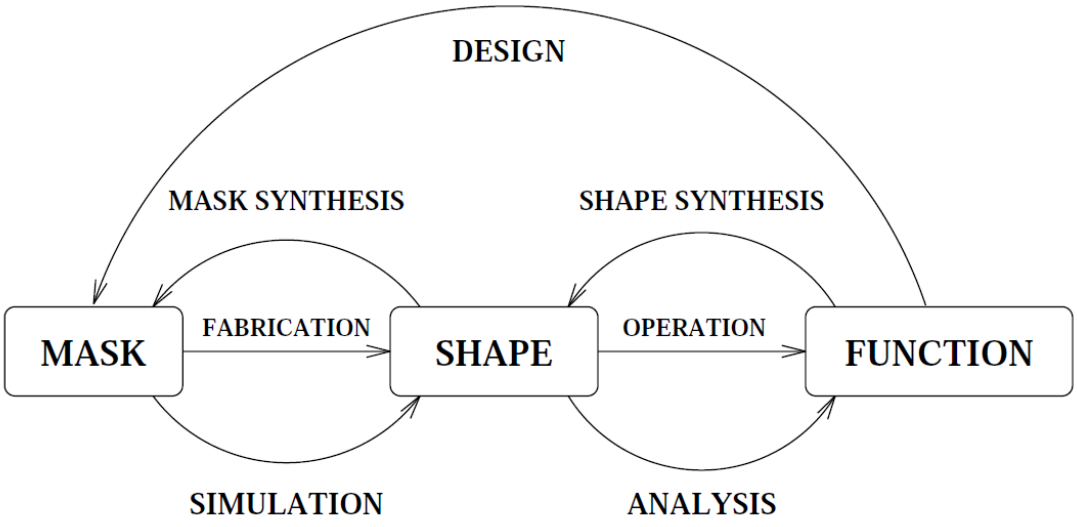


Figure 1.1 Early concept of MEMS design process, taken from [21].

A more recent attempt to visual the design process for MEMS was created by Senturia [16] and is shown in figure 1.2. This more abstract definition looks to encapsulate all levels of design in which user information can be applied in a hierarchical 'top-down' or bottom-up' manner. In both examples the emphasis is placed upon the modelling and 'simulation' of the MEMS device, a key goal outlined in the report [21] and important to the overall aim of developing automated design synthesis tools and methodologies.

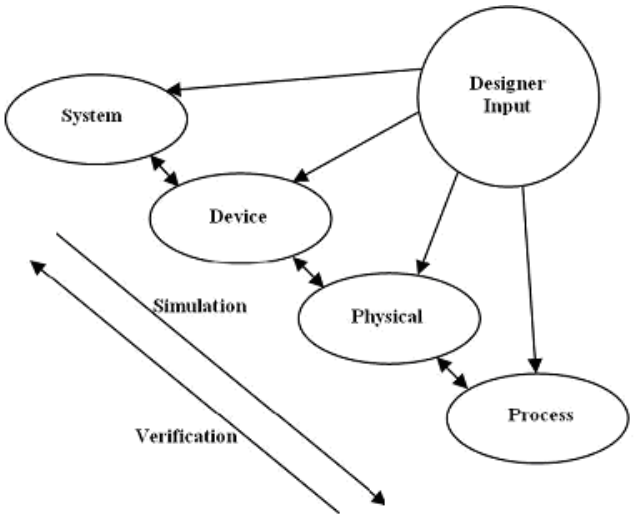


Figure 1.2 Four design levels for microsystems, as identified in [16]

A closing statement for the workshop concluded with the remarks 'The time is now ripe to define and develop structured design methods and to take advantage of the still formative nature of the field' [21]. In some ways this has been achieved, as there has been large success in creating ever more accurate modelling and analysis software over a wide variety of design levels and in facilitating communication between these levels. However this success has not been equal when it comes to any MEMS design methodology in particular in an automated fashion.

In recent years there have been an interest in the standardising of the MEMS design process and although a general outline exists there are no methodologies which define the design flow to achieve unambiguous design targets [22]. These outlines also still fall short of describing how these levels can be connected or how best to take advantage of the number of model abstractions available to the designer. A large part of this is that anyone looking into developing a framework or methodology into a structured design process of MEMS has to consider a number of characteristics such as coupled physical domains, different simulators, design tools and their environments, different levels of abstraction and how to communicate between these levels [23]. Even as recently as 2008 the integration of MEMS into a standard design flow remains open [24].

Two standard approaches a MEMS designer can use to synthesis of MEMS device in particular with regards to modelling and simulation are the bottom up and top down methods, linked to the hierarchy of design shown in figures 1.1 and 1.2. The standard bottom up approach to MEMS design begins with a group of mask layouts and process information that is used to build a 3D geometry which can then be analyzed dynamically for its performance on some function. The design is then evaluated, and a decision is made as to whether some stopping design criteria are

met or whether to jump back to the beginning and rewrite the fabrication information for a new design. This approach has several disadvantages, firstly thinking of design in the context of mask layouts and process information does not fit well with the normal fixation of shapes, functions and component design that most design engineers face. Secondly it is uncommon for a designer to have intricate knowledge of the methods applicable to fabrication and mask layout design, and this particular level of design does not fit well with both automation and optimization techniques.

In contrast a top-down approach, focusing on building designs starting from simple 'behavioural models' and then using this information to build ever more complex layout or 3D models more closely mimics the approach found in digital VLSI design. The movement into a top-down MEMS design methodology has enabled a dramatic increase in design productivity, measured in an increase of manageable design complexity, a decrease in both the time spent working on design and the number of errors that may come about as a result [25-27]. In the following paper [25] it was noted that "A hierarchical 'top-down' design flow starting from system concept, to components, to low-level functional elements is necessary to drive the design process directly with desired application specifications".

Another advantage in developing a successful design methodology is that it might break or alleviate the dominance of high volume production devices. Currently devices such as automotive accelerometers, gyroscopes and pressure sensors form the bulk of products that are developed, while custom built though potentially successful devices require specialist designers, with intricate domain and application knowledge [28]. The creation of a design optimisation strategy could help remove boundaries associated with such custom and often complex devices by looking to improve automation and optimisation of their design.

At present there are a number of design and simulation tools available, each with their own level of modelling abstraction and simulation accuracy. These can all be linked in some way to the hierarchical levels of design described in figure 1.2 and available to a designer for MEMS design synthesis. The current approaches to automated MEMS design synthesis focus upon single levels of modelling and analysis often tied to a single model or discipline. This can be disadvantageous as it can restrict the number of tools available to a designer during design optimisation and synthesis. MEMS are often complex consisting of many individual components and devices often coupled together and coordinating functions to meet some goal set out by the designer. The ability to capture all this information into a single model or representation can be difficult and lead to problems during synthesis and optimisation of the device for the more traditional automated optimisation routines currently used.

Though current advances in both modelling and analysis of MEMS and their automated design synthesis and optimisation have moved on through the recent decades there are still gaps in how to best utilise all levels of design in a structured methodology. The next stage is to look into how a design optimisation strategy can be applied which looks to take advantage of all levels of design rather than as is currently the normal practice just one.

One successful approach to design optimisation for hierarchically structured systems is found in the maxim of multi-level design. Multi-level design covers a number of successful strategies from multidisciplinary optimisation to distributed hierarchical design optimisation and in both areas

has shown substantial improvements in both improved design and cost reduction of design. These approaches have shown a marked improvement in the speed up of the design optimisation process, the ability to simplify the design search space and handle complex multidisciplinary optimisation problems.

Multidisciplinary optimisation (MDO) has been applied to many engineering problems however the use of population based MDO less so, and in the guise of MEMS design synthesis is virtually nonexistent. MEMS design synthesis is highly multidisciplinary however it brings with it certain problems which need to be addressed when looking to apply MDO and develop an MDO algorithm capable of undertaking design synthesis and optimisation. Still the field of MDO offers a promising avenue for improvement of the MEMS design process.

1.3 Research aim, objectives and approach

1.3.1 Research aim and objectives

The aim of this thesis lies in the construction, experimentation and analysis of both multi-level and multidisciplinary optimisation strategies towards MEMS design synthesis, with a goal to evaluate the benefits and drawbacks such methodologies can bring to the field of MEMS design. Such an aim looks to push forward and improve the current state of the art in automated MEMS design synthesis and on a broader level gain further understanding in the applicability and performance of multi-level and multidisciplinary design optimisation strategies towards engineering design.

In order to achieve the goal set out a number of research objectives were pursued in this thesis and are described as follows: (1) To undertake a study of literature in the multi-objective, (Chapter 2) multi-level and multidisciplinary design synthesis and optimisation of MEMS (Chapter 3); (2) To develop multi-level strategies for application to automated MEMS design synthesis and optimisation (Chapter 4); (3) To outline and develop a multidisciplinary optimisation algorithm for application to automated MEMS design synthesis and optimisation (Chapter 4); (4) To develop a framework for automated multi-level and multidisciplinary optimisation of MEMS (Chapter 4); (5) To construct a hierarchical MEMS bandpass filter benchmark case study (Chapter 5); (6) To undertake multi-level and multidisciplinary optimisation strategies for automated MEMS design synthesis (Chapters 6, 7, and 8). The detailed motivation and aim of each research objective are described in the following subsections.

1.3.1.1 Research objective 1: To undertake a study of literature in the multi-objective, multi-level and multidisciplinary design synthesis and optimisation of MEMS.

Traditionally the design synthesis and optimisation of MEMS has been hand driven. This process often revolves around the design, modelling and parameterization of a device, before an iterative methodology sees the designer input various sets of design variables in the hope of optimising the device. The outcome is often a trial and error approach where fabrication leads to testing and ultimately re-evaluation and redesign, in what is both time consuming and expensive.

Over time automated methods for MEMS design synthesis and optimisation have been incorporated into the design process, traditionally using numerical methods, such as local gradient-based search algorithms [29][30]. However the complexity of MEMS device performance and their design search space landscape mean local search methods such as these will struggle and as a result more powerful stochastic methods from the field of evolutionary computation have been incorporated.

The first research objective focuses firstly on this latest addition to automated MEMS design synthesis and optimisation, the application of evolutionary algorithms, in particular given the engineering nature, multi-objective evolutionary algorithms. There are a number of examples of their successful application, particularly those hailing from the BEST group from the University of California [31][32][33], however the multi-objective evolutionary algorithms (MOEAs) used are no longer state of the art. Therefore we look to extend this work by creating a platform for automated MEMS design synthesis and then using the most referenced state of the art MOEAs within MEMS literature re-evaluate their performance on a number of MEMS design problems. In tandem with this a critical analysis of MEMS design synthesis with particular focus on automated techniques is also undertaken.

The process of MEMS design optimisation can also be broken down into a number of hierarchical levels into which a designer may provide input and ultimately model, analyse and optimise a device. However the majority of automated design synthesis methodologies focus upon only a single level or discipline when looking to synthesize or optimise a MEMS device.

The systems or devices a designer may wish to create may also contain many components all coupled together to function as whole and directed to provide some designed behaviour. The complexity and dimension of the search space can be large especially given the large number of disciplinary analyses and parameterized design variables associated with the system or device. Two areas of research, the field of multi-level and multidisciplinary design optimisation have developed to overcome some of the challenges of hierarchy and design complexity found in many engineering disciplines.

Therefore research objective one extends into the field of multi-level and multidisciplinary design by reviewing and critically analysing the best strategies to apply towards automated MEMS design synthesis. This is in relation to the hierarchical levels present in the MEMS design process and the available modelling and analysis tools.

The goal of Research Objective 1 is to critically evaluate the current state of the art in MEMS design synthesis and optimisation. This is finalized in the development of a platform for automated MEMS design synthesis and optimisation through the coupling of modelling / analysis tools to the stochastic multi-objective optimization algorithms within evolutionary computation. A number of MEMS design problems from the literature are than used to evaluate the role MOEAs have in the area of MEMS design synthesis and optimisation. Finally the field of multi-level and multidisciplinary design optimisation is reviewed and critically analysed for the best strategies to apply towards automated MEMS design synthesis.

1.3.1.2 Research objective 2: To develop multi-level strategies for application to automated MEMS design synthesis and optimisation.

Multi-level strategies have been successful in other engineering fields providing reductions in computational cost [34][35] through there interaction with numerous levels of modelling abstraction and have been employed to reduce the complexity of large engineering problems consisting of many objectives, constraints and design variables [36][37]. They have also provided solutions to engineering problems which require high levels of refinement due to their scale and sensitivity to variation, a common theme in MEMS devices with their non linear behaviour [38][39].

Having critically reviewed and analysed the best strategies to apply towards automated MEMS design synthesis the next objective looks to fully develop them into a number of MEMS multi-level design optimisation strategies to be applied through the remainder of this thesis.

The goal of Research Objective 2 is to develop multi-level design optimisation strategies suitable for application to the field of automated MEMS design synthesis and optimisation.

1.3.1.3 Research objective 3: To outline and develop a multidisciplinary optimisation algorithm for application to automated MEMS design synthesis and optimisation.

The field of MEMS is inherently multidisciplinary and now covers a wider range of disciplines than the simple mechanical and electrical domains. Often the complexity of MEMS design can stem from the fact that they consist of multiple devices or components often from different disciplines all linked and coordinating together to produce a desired function or behaviour.

Coordination in the design of the various components can be difficult for a designer as altering the attributes of one component may have inadvertent effects on another. A class of coordination algorithms are found within the field of multidisciplinary optimisation and tasked with solving problems which exhibit strong coupling through their disciplines or component units and are difficult to solve as a single integrated system. The third research objective is therefore concerned with the field of MDO algorithms and how they can best be applied to aid MEMS design synthesis and optimisation.

The current state of the art in MDO coordination algorithms typically focus upon single solution structures utilizing local search algorithms such as gradient based search or sequential quadratic programming (SQP) [40]. Population based MDO algorithms are however rare, and in application to MEMS design nonexistent. In a standalone population based approach to design optimisation such as genetic algorithms (GAs) the design problem can be readily constructed as an all-in-one formulation with all decision variables, objectives and constraints of the system represented together. In a population-based MDO algorithm this is more difficult particularly when the design problem is decomposed into separate disciplines and / or component objects.

Therefore in addition to researching the applicability of MDO to MEMS design, the construction of a novel population based MDO algorithm towards MEMS design synthesis and optimisation is also undertaken.

The goal of Research Objective 3 is to evaluate the field of multidisciplinary optimisation algorithms and develop a novel MDO algorithm for application to automated MEMS design synthesis and optimisation.

1.3.1.4 Research objective 4: To develop a framework for automated multi-level and multidisciplinary optimisation of MEMS.

In order to successfully apply both multi-level and multidisciplinary optimisation algorithms to the design synthesis and optimisation of MEMS a computational design optimisation framework needs to be constructed. In the past a number of contributions to the design process have been included, with a move to stochastic multi-objective genetic algorithms [41], component-based representations [42], interactive design [33], and more complex hybrid global and local search algorithms [43] to name a few. Any framework built should include or be flexible enough to accommodate such contributions so as to ensure the MEMS design process remains moving forward.

The addition of multi-level and multidisciplinary optimisation to the design process of MEMS brings with it its own challenges. The ability to handle multiple modelling and analysis tools, decomposable representations and multiple optimisation routines are but a few of these. The fourth research objective is tasked with building a computational framework for the automated multi-level and multidisciplinary optimisation of MEMS. All algorithms developed on the platform towards this goal also need to be validated against known examples within the literature to assess their suitability for evaluating the application of multi-level and multidisciplinary optimisation to the MEMS design process.

The goal of Research Objective 4 is to develop a framework to build the multi-level and multidisciplinary optimisation strategies outlined for application to automated MEMS design synthesis and optimisation.

1.3.1.5 Research objective 5: To construct a hierarchical MEMS bandpass filter benchmark case study.

The literature for MEMS design synthesis and optimisation is littered with examples that predominately feature single devices / components or disciplines. However design problems which cover modelling and analysis from more than one level of design are rare, in particular when concerned with automated design methodologies. Hierarchical design optimisation, that is design optimisation linked to the design levels outlined by Senturia [16] (System, Device, Physical, Process) and that involve exploiting more than one level of design, to the authors best knowledge, are practically non-existent in the design problems present within the literature. There are some examples where the overall design goal involves the design synthesis and optimisation of a system which contains multiple levels of modelling and analysis, however these involve optimisation in isolation with each level of design done separately and un-coupled.

The field of MDO also lacks suitable benchmark problems with the majority either purely mathematical without an engineering origin or irreproducible [40].

In order to undertake an evaluation into the performance of multi-level and multidisciplinary optimisation strategies towards automated MEMS design synthesis and optimisation new design

problems need to be constructed. This is predominately the main aspect of research objective 5 with the creation of a multi-level and multidisciplinary benchmark problem which covers a number of the hierarchical levels of MEMS design. The benchmark problem should where ever possible be taken from examples from the literature where their modelling and analysis have been suitably peer-reviewed and where possible the device or component fabricated.

The goal of Research Objective 5 is to develop a multi-level and multidisciplinary optimisation benchmark problem in order to provide a suitable test bed for the current state of the art single level and the proposed multi-level and multidisciplinary optimisation strategies.

1.3.1.6 Research objective 6: To undertake multi-level and multidisciplinary optimisation strategies for automated MEMS design synthesis.

The last objective covers the bulk of the research by applying the multi-level and multidisciplinary optimisation strategies built on the developed framework on a series of benchmark case studies created to cover a number of the hierarchical levels of MEMS design.

This is compared against the current state of the art in MEMS design, a single level or all-in-one methodology utilizing multi-objective genetic algorithms from the field of evolutionary computation. Both sets of strategies are applied to examples which cover one, two or three levels of the hierarchical design process, each containing their own modelling and analysis tools common to MEMS designers.

Analysis of the results is than given on the benefits and drawbacks in applying such strategies to the design process of MEMS, recommendations on which strategies and what circumstances should they be applied to design synthesis and optimisation are also discussed.

The goal of Research Objective 6 concerns itself with a comparative analysis of the benefits and drawbacks of multi-level and multidisciplinary optimisation to the current state of the art in automated MEMS design optimisation.

1.3.2 Overview of research methodology

In order to evaluate the role multi-level and multidisciplinary optimisation can play in automated MEMS design synthesis, a number of objectives need to be addressed. Over the last decade there has been a marked increase in automated design synthesis of MEMS, all designed with the aim to facilitate more rapid production and lower costs throughout the design process. The current state of the art in automated MEMS design optimisation has moved towards the use of more stochastic rather than traditional numerical methods to overcome some of the difficulties in MEMS design. This work, the bulk of which has been undertaken by the BEST group from the University of California involves the use of algorithms from the field of evolutionary computation, in particular given the engineering nature of MEMS design, multi-objective evolutionary algorithms.

Therefore in order to address the first objective, the first task is to undertake a critical literature review of the design process of MEMS with a particular focus upon automated methodologies and tools. This information is then used to produce a design optimisation platform which links the field of evolutionary computation to the modelling and analysis tools of MEMS design. This

platform can then be used to perform an up-to-date evaluation of the performance of multi-objective evolutionary algorithms on MEMS design synthesis and optimisation using the current state of the art MOEAs. The evaluation is both qualitative and quantitative through the use of known multi-objective evaluation metrics (hypervolume [148][149]) and a comparison to the latest state of the art results found in the literature. This provides a solid foundation of the current state of the art in automated MEMS design optimisation and the correct routines for linking modelling and analysis tools to optimization algorithms.

The next step involves a critical literature review of the current state of the art in multi-level and multidisciplinary optimisation with a focus upon engineering and real world examples. The information gained can then be used to outline a number of multi-level strategies applicable to MEMS design synthesis and optimisation. These strategies are chosen based upon the current state of the art in multi-level design and their suitability to the hierarchical process of MEMS design optimisation and synthesis and are later developed to address research objective two.

The outcome of the critical literature review for the current state of the art in multidisciplinary optimisation is a focus upon population based MDO strategies particularly of a multi-objective nature with an outline for a new novel approach to MDO later developed to address research objective three. The end results of both methodological processes are the evaluation and construction of the multi-level and multidisciplinary optimisation strategies employed within this thesis for MEMS design.

Having established a design optimisation platform routine and outlined the various multi-level and multidisciplinary strategies the next step is to address research objective four with the construction of a framework suitable for automated multi-level and multidisciplinary optimisation of MEMS. Though there are a number of optimisation libraries and frameworks for evolutionary computation available within the literature, they are often linked to the standard routines of an all-in-one design process. Both multi-level and multidisciplinary optimisation strategies require more complex hierarchical structures, consisting of multiple populations and multiple optimisation routines.

An object-orientated computational framework is therefore implemented in the C# .Net programming language that is based upon a structured and coordinated modular system that allows the decomposition of common processes into a series of linked modules to address research objective four. The current state of the art multi-objective genetic algorithms SPEA2 [132] and NSGAII [45] are built within the system and then used as a basis for the more structurally complex multi-level and multidisciplinary optimisation algorithms outlined previously. Both multi-objective algorithms, along with the multidisciplinary optimisation algorithm are validated against known problems within the literature and their evaluation is based upon known multi-objective metrics (hypervolume [148][149]) to insure their accurate implementation within the developed framework.

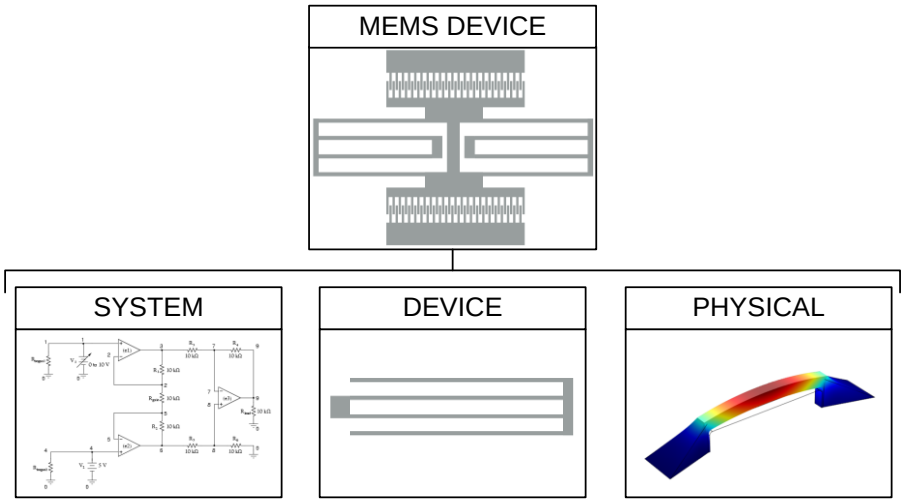


Figure 1.3 MEMS hierarchical case study

The next step within this research is the construction of a suitable MEMS hierarchical case study that can be used as a template for undertaking an evaluation into the performance of multi-level and multidisciplinary optimisation strategies for MEMS design synthesis. In order to address research objective five a MEMS bandpass filter case study that encompassed three levels of the MEMS design hierarchy was constructed and evaluated. The chosen hierarchical levels are based upon those outlined by Senturia [16] (System, Device and Physical) and each involve their own set of unique models and analysis tools. This involved the integration of a number of analysis tools into the computational framework followed by a series of experimentation using a series of MEMS bandpass filter designs. The results are compared against the current state of the art in automated MEMS bandpass filter design and show good agreement in terms of performance.

Next in order to address research objective six a comparative investigation between the current state of the art in automated MEMS design optimisation and the developed multi-level and multidisciplinary optimisation is undertaken with detailed analysis on the benefits and drawbacks of both methodologies. This is structured into three sections of research focusing upon first a look into multi-level and multidisciplinary design optimisation at a single 'Uni' level of modelling and analysis at each of the hierarchical levels outlined (System, Device and Physical). Next follows coupled 'Bi' level design problems (System – Device and Device – Physical) exploring more complex MEMS design problems which span more than a single level of modelling and analysis. Finally a 'Tri' level design problem which contains modelling and analysis from all three levels (System – Device – Physical) is constructed and utilised to test each strategy. In all three sets of experimentation both qualitative and quantitative analysis is undertaken using known multi-objective evaluation metrics (hypervolume [148][149]) to compare and contrast the current state of the art single level MOGA strategy with the proposed multi-level and multidisciplinary strategies.

1.4 Outline of the thesis

The organization of this thesis is shown in figure 1.4 and covers all chapters and their major contributions to this research as a whole. Chapter 2 presents an overview of the modelling, analysis and optimisation of MEMS with a focus upon automated methodologies. Next follows

construction of an automated design optimisation platform to validate the efficacy of multi-objective genetic algorithms over a number of MEMS design problems; Chapter 3 concerns itself with the fields of multi-level and multidisciplinary optimisation. A critical review of the literature on multi-level design optimisation is undertaken and a number of strategies are selected for application to MEMS. In addition the literature for the field of multidisciplinary optimisation is critically analysed with a focus upon population-based multidisciplinary optimisation. A novel multidisciplinary optimisation algorithms for application to MEMS design is then outlined; Chapter 4 outlines the design strategies and computational framework that will be built and used to house the multi-level and multidisciplinary optimisation algorithms for application to MEMS design synthesis and optimisation; In Chapter 5 a hierarchical MEMS case study benchmark design problem is constructed as a basis for evaluating the performance of the multi-level and multidisciplinary optimisation strategies. The case study is based upon known MEMS models taken from the literature and tied to modelling and analysis tools available and integrated into the computational framework; The Chapters 6, 7 and 8 contain a number of comparative experiments between the outlined multi-level and multidisciplinary optimisation strategies to the current state of the art in automated MEMS design synthesis and optimisation. Finally chapter 9 concludes with a discussion of the overall impact both multi-level and multidisciplinary optimisation has had on the results and MEMS design overall, followed by a look at some recommendations for future research.

1.5 Summary

This chapter has summarised the background of MEMS and defined the problem statement that this thesis is looking to address. From this a number of aims and research objectives were outlined along with an overview of the research methodology employed to accomplish them. Finally an outline of the thesis and each individual chapter is given, along with how they link to each of the objectives outlined.

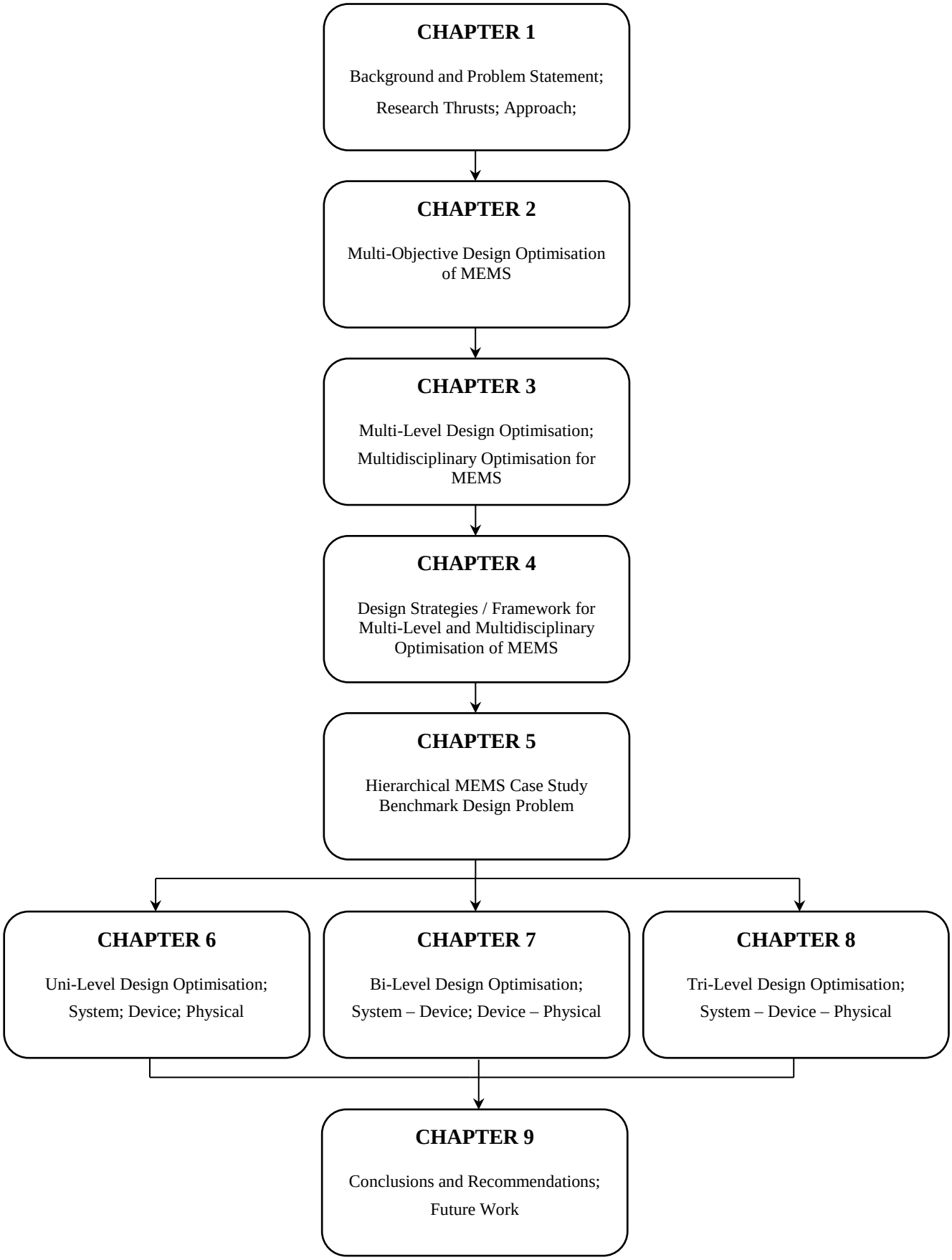


Figure 1.4 Organization of thesis

2

Literature Review on Design Synthesis and Optimisation of Microelectromechanical Systems

This chapter presents the current state of the art in MEMS design optimisation through a look at the modelling and analysis methods and tools available to designers during the MEMS design process. Both traditional and more current state of the art automated design synthesis and optimisation methodologies are discussed with a particular focus upon the field of evolutionary computation. A multi-objective design synthesis and optimisation platform is constructed to evaluate the efficacy and role multi-objective genetic algorithms can play in MEMS design over five unique case studies identified within the literature.

2.1 Modelling and analysis of microelectromechanical systems

Designing, fabricating and testing MEMS in an iterative fashion with the goal to produce an optimal device is both time-consuming and expensive. This is somewhat a result of the reliance on an old “back of the envelope” [43] strategy, where a “build and break” iterative process is employed towards the design, fabrication, testing and often consequently redesign of a device. Therefore an important part of current MEMS design processes is the integration of modelling and analysis tools to reduce the time and cost of manufacturing a device.

Beginning in the late 80’s and early 90’s a host of computer-aided design and simulation tools were constructed for the specific purpose of stimulating growth and design of MEMS. The modelling of microelectromechanical devices can be broken down into three basic tasks, the creation of a model representation of the device, the simulation of this model based on physical behaviour associated with it and finally the visualization and analysis of the simulation event.

A number of modelling and analysis tools are now available to designers, each with their own level of computational cost, accuracy and disciplinary domain. The arrangement of these tools has been outlined by Senturia [16] in which the overall design process is broken up into a hierarchy of levels (System, Device, Physical and Process). Here the system level is often where there is an integration of the mechanical with the electrical components of the device, often representing the most abstract of modelling and analysis tools. The physical and device level often contain physical and realisable representations of the device of varying granularity at a

cost of increased simulation and analysis time. The process level itself stands out from the others as it often deals with the process and fabrication information needed to build the designed device rather than being used as a modelling and analysis level for device design on its own. This hierarchical nature of the MEMS design process provides designers with the problem of how best to approach the possible decomposition of the device at the various levels of modelling and analysis abstractions available to them, particularly as each level of abstraction comes with its own benefits and tradeoffs.

The modelling and analysis levels outlined by Senturia [16] are discussed and analysed in detail in the following sections, along with examples of their application to design optimisation in the literature. For further review of the modelling and analysis of MEMS see [47] and [48].

2.1.1 System level modelling and analysis

The system level of modelling and analysis is positioned at the top in what is often the highest and most abstract level of modelling representation. Where other tools available to a MEMS designer offer more functionality or accuracy in physical behaviour, they often come with a large computational cost which is exhibited in long simulation and analysis times. The system level provides designers the ability to build simplified low cost models that can encompass the whole system, mechanical and electrical, in one integrated representation. These schematic representations often allow designers to rapidly evaluate many alternative designs and build intuition on how the overall system will perform or behave [49].

The modelling and analysis of the system level focuses upon the use of lumped element circuit models [16][50], block diagrams [51] and more recently bond graphs [52][53] to model device performance, utilising powerful circuit simulators. They also provide the possibility to interface with the mechanical elements of the device, either through analytical models, hardware description language (HDL) models, reduced order models or alternatively electrical equivalent representations of the mechanical component.

The use of block diagram tools such as Matlab-Simulink [51] have a clear input-output definition similar to that found within integrated circuit schematic design tools with information flowing through the blocks. The behaviour of components or whole devices can be modelled relative to the parameters of its design and these can then become interconnected to form whole or even more complex systems. The disadvantages of such a system lie within the lack of a conservation of energy.

A modelling approach which overcomes this is found in the domain of lumped element modelling, which tries to capture the behaviour of devices or components often using a hardware description language in the form of ordinary differential equations that offer more fidelity in the accuracy of the device. The advantage of conservation of energy within the modelling representation is the ability to easily interface with the electrical components of the device [54].

A new system level modelling application to MEMS is that of bond graphs [52] which provide a more adaptable and promising approach to the modelling of multi domain dynamic systems such as MEMS. Bond graph models under the guise of the conservation of energy are able to

describe the dynamic behaviour of physical systems by the connection of idealized lumped elements [53]. The benefit of such a modelling tool is its ability to model not just the mechanical and electrical domains but also hydraulic, pneumatic components in a unified system [53].

The disadvantage of system level tools is that they often do not capture the layout of the device and are restricted to higher generalizations of the devices they model, for example rather than the physical geometry of a MEMS bandpass filter, only electrical equivalent values for mass, stiffness or damping are modelled [50]. There is also a high level of expert knowledge that is often required to build such system level models and changes in device topology or geometry may require new models to have to be built. The modelling tools themselves are also restricted from analyses common to solid models such as stress and fluid dynamics that are themselves associated with the more computational expensive levels of modelling and analysis. Table 2.1 provides a summary of some of the major developments in system level modelling and analysis.

Table 2.1 Chronological review of developments in MEMS system level modelling and analysis

Contributors	Date	Methods	Applications	Description
Nagel, L. and Pederson, D.O. [55]	1973	Behavioral simulator	Numerous	SPICE is an integrated circuit simulation package
Verilog-AMS [56]	1984	HDL simulator	Numerous	Verilog-AMS is a hardware description language tool
ELDO [57]	1987	Behavioral simulator	Numerous	ELDO is an integrated circuit simulator package
Spectre [58]	1988	Behavioral simulator	Numerous	Spectre is an integrated circuit simulation package
Matlab-Simulink [51]	1994	Model-based simulator	Numerous	Commercial simulation package which utilizes block diagrams for MEMS device analysis
Broenink, F. [59]	1999	Bond graphs	Numerous	Bond graphs and their application to physical systems
Getreu. I. [60] and Chwirka, S. [61]	1989, 2000	HDL simulator	Numerous	Saber is a mixed-signal analysis tool for complex mechatronic systems

2.1.2 Device level modelling and analysis

The device level holds some of the most common methods used for modelling and analysis in MEMS design. The level is associated with relatively low cost simulation however unlike the system level there is a move into the realisation of the device through 2D layout modelling representations.

Where system level tools provide suitable models and analysis for conceptual design and exploration of alternative solutions to the design problem faced by the designer the device level provides modelling and analysis for the realisation and layout of the system at a reasonable computation cost.

The device level can be partitioned into three separate methods of modelling and analysis, macromodelling, nodal modelling and reduced order modelling. Each of these techniques is

discussed in the following sections along with examples from the literature on their application in MEMS design.

Macromodelling

Analytical or semi-analytical macromodelling is currently the most common approach to develop a model to analyse device behaviour. Physical models are commonly constructed from ordinary differential equations (ODEs) and are valued for the insight they can bring in terms of device behaviour and performance and their link to geometric parameters and material properties [16].

In order to generate such models there is often a need for a high level of expert knowledge on the device itself in order to create models that contain all the necessary physics and interactions. The creation of the model can be as a result time consuming and unfortunately it is generally not possible to build upon such models to create larger more complex systems. An example would be the modelling a single MEMS resonator device, a key component of a MEMS bandpass filter, cannot later be composed into a multi MEMS resonator system and instead a new model needs to be created [49]. Tools designed to aid in macromodelling creation have also been developed which look to automatically build macromodels from provided solid model representations [62].

There is also the problem with how best to undertake parameter estimation for the model itself. Numerical simulation is favoured over analytical techniques due to the ability to determine the explicit dependence of model parameters on the design attributes [49] and as a result a number of analytical and numerical hybrid tools have been developed to aid such modelling [63] [64].

Other frailties that lie with macromodelling are the lack of a standard method for generating models which can often mean their accuracy needs to be validated against experimental results or detailed simulation, and even if this is achieved the models are only as accurate as the designer who builds them [49].

Detailed examples of macromodel creation can be found in [49] focusing on a single MEMS resonator and in [65] with the development of a triaxial microaccelerometer. Further examples are also shown in table 2.2 which provides a summary of the major developments in device level modelling and analysis.

Nodal modelling

In a similar vain to electronic circuit design, tools that consist of composable models for a number of MEMS building blocks have come about. Nodal analysis usually involves the solving of coupled nonlinear differential equations widely used in circuit analysis such as SPICE [55]. The approach decomposes the circuit into N-terminal devices or atomic elements each of which is modelled by ordinary differential equations with the coefficients parameterized by device geometry, and material properties derived from real data or process specifications [66]. Each of these elements can then be combined together at their terminals, or nodes to create larger more complex devices which can then be solved using nodal analysis [67].

The two major contributions to nodal analysis and its application to MEMS design are the tools NODAS [68][69] and Sugar [66]. The development of NODAS derived from the ‘Nodal Design of Actuators and Sensors’ in [69] was motivated by the need to overcome constraints of previous

behavioural modelling which focused on individual devices as a whole. The new tool allowed designers to build devices from a hierarchical set of atomic elements and construct planar MEMS devices consisting of novel geometric and topological settings. NODAS is developed using the Saber [60] tool with each lumped model element written in the MAST hardware description language [70].

Another nodal analysis tool developed in parallel is Sugar [66] developed within the University of California, Berkeley. In homage to a previous incarnation of the developed SPICE circuit analysis tool, Zhou et al. [66] looked to build a VLSI inspired approach for MEMS design modelling and analysis. Once again a number of atomic elements written for the platform Matlab [51] are available to a designer to build larger connected devices, with Matlab providing easy accessibility and graphical display of modelling and analysis.

Both NODAS and Sugar show strong affinity to analysis results from the more accurate FEA methods [71][72] though at a significant reduction in computational cost, however similar to system level models the nodal tools cannot provide some complex analyses such as stress calculation or fluid dynamics required by some devices.

Nodal analysis has also found itself incorporated into a number of tools which contain multiple levels of modelling and analysis for example Architect [73] or Intellisuite [18]. A number of nodal modelling and analysis example are shown in table 2.2.

Reduced order modelling

Model-order reduction (MOR) strategies have been developed to extract nonlinear macromodels from devices that exhibit strong nonlinear and dynamic behaviours [49]. MEMS are multidisciplinary and often exhibit nonlinear behaviour and many interacting forces over a number of devices and components. Such system are only going to get larger and more complex and simulating such systems using expensive finite element methods (FEM) to capture their behaviour simply becomes more intractable. As a result a number of methods looking to reduce the order of complexity while still retaining a high level of accuracy of MEMS modelling and analysis have been developed.

These strategies look to create reduced order models by using quadratic techniques [74][75][76], or an Arnoldi algorithm [77][78] to reduce the finite element or finite difference method (FDM) formulations of sold model representations [47]. However such approaches are only appropriate for linear or weakly non-linear systems as they can be incapable of capturing the dynamical behaviour of nonlinear models [47]. To overcome this Rewienski and White [79] proposed an algorithm based upon a trajectory piecewise-linear model order reduction for tackling highly nonlinear systems.

Other approaches for generating reduced order models are the Karhunen-Loeve/Galerkin method [80], which builds basis functions from an ensemble of FEM data that can suitably characterize the operating conditions of the device. In this instance, an example of a pressure sensor was used, a highly non-linear device, with results showing good agreement with traditional finite difference methods but also a marked speedup on simulation time. An

overview of a reduced order model method that utilizes a Krylov-subspace can also be found in [49] and [81] respectively.

The downside is that such models are built from existing models and therefore are suitable for simulation but not design synthesis of the geometry and topology of a device. Also some methods require numerous expensive simulation and analysis to build the required data of the device in order to extract the information needed to build reduced order models, an example being the basis functions required for the Karhunen-Loeve/Galerkin method [47]. Examples of model-order reduction strategies applied to MEMS are included in table 2.2.

Table 2.2 Chronological review of developments in MEMS device level modelling and analysis

Contributors	Date	Methods	Applications	Description
Lee, K. W., and Wise, K. D. [63]	1982	Analytical hybrid	Pressure sensor	SENSIM, simulation tool for the calculating the output response of piezoresistive or capacitive pressure sensors
Bin, T. Y., and Huang, R. S. [64]	1987	Analytical hybrid	Pressure sensor	CAPPS, a simulation tool for pressure sensor analysis and geometrical design
Vandemeer, J., et al. [68]	1997	Nodal	Crab-leg resonator, electrostatic actuator	A nodal simulator that combines atomic element building block behaviors into higher order structures for device level analysis
Wang, F., and White, J. [77]	1998	MOR	Numerous	Model order reduction of MEMS devices using the Arnoldi method
Swart, N. R., et al. [62]	1998	Analytical	ADXL76 accelerometer	AutoMM, is a CAD tool for the automatic creation of macromodels
Vandemeer, J., et al. [69]	1998	Nodal	Crab-leg resonator, microgyroscope, single beam cantilever	Extending previous work [68] to incorporate actuators and sensors into simulation software coined NODAS
Zhou et al. [66]	1998	Nodal	Resonators, planar devices	A nodal simulator inspired by VLSI and utilizing atomic elements for the construction of planar MEMS
Bechtold, T., et al. [74]	2003	MOR	Microthruster	Electro-thermal model order reduction for a microthruster
Zhu, M., et al. [65]	2004	Analytical	Triaxial accelerometer	Formulation of a dynamic model for a triaxial accelerometer derived using lagrange equations
Ying, Y. J., and Yu, C. C. [76]	2004	MOR	Microactuator, infrared imager	Model order reduction technique for generating heat-transfer macromodels

2.1.3 Physical level modelling and analysis

The most computational expensive of all modelling techniques are found generally within the physical level and the focus upon solid 3D modelling and analysis. Where analytical and macro modelling are more common within the literature of MEMS design, it is to numerical modelling

techniques which designers often look to finalise designs and verify their function and behaviour.

Table 2.3 Chronological review of developments in MEMS physical level modelling and analysis

Contributors	Date	Methods	Applications	Description
Puers, B., et al. [82]	1989	Numerical	Capacitive pressure sensors	CAPSIM, finite element software for capacitive pressure sensor anlysis
Crary, S., and Zhang, Y. [83][84]	1990, 1991	Numerical, process modeler	Pressure sensor	CAEMEMS, combining process modeling, mesh creation and FEA into a CAD environment
Zhang, Y. et al. [85]	1991	Numerical, process modeler	Pressure sensor	Early example of cascade from mask layout and process information simulation to 3D model and FEA analysis
Senturia, S. D., et al. [20]	1992	Numerical, process modeler	Square diaphragm capacitance and beam microstructure	MEMCAD, a package combining process modeling and simulation with simulation analysis tools
Folkner, B., et al. [86]	1992	Numerical	Capacitive and piezoresistive sensors	Simulation tool for the design and optimization of sensor devices
Solidworks [87]	1995	Numerical	Solid modeler	CAD tool for the 3D visualization and FEA of MEMS devices
Funk, J. M., et al. [88]	1998	Numerical	Microactuator	SOLIDIS, a analysis and simulation tool for encompassing electrothermal,surface-electrostatic and piezoelectric effects
Architect [73]	2008	Numerical, mixed	Numerous	Architect is a mixed simulation package for MEMS
ANSYS [89]	2008	Numerical	Solid modeler	CAD tool for the 3D visualization and FEA of MEMS devices
ALGOR [90]	2008	Numerical	Solid modeler	CAD tool for the 3D visualization and FEA of MEMS devices
ABAQUS [71]	2008	Numerical	Solid modeler	CAD tool for the 3D visualization and FEA of MEMS devices
MEMSPRO [19]	2008	Numerical, mixed	Numerous	Mixed CAD tool for MEMS devices with process modelling, schematic capture and model generation
COMSOL [91]	2008	Numerical	Solid modeler	CAD tool for the 3D visualization and FEA of MEMS devices

Typically this involves the use of finite element, finite difference, or boundary element methods (BEM) to find approximate solutions to partial differential equations (PDEs). Often this involves the construction of 3D solid models of the device that is to be analysed along with any additional boundary constraints, with the model discretized into a set of elements described by these PDEs. The information held within each element can be used to describe or calculate responses to a number of physical or disciplinary phenomena inside an object or structure. This is advantageous as more accurate and multidisciplinary analysis becomes available, such as fluid dynamics and thermal stress analyses not often available at more abstract modelling levels. Other advantages include the ability to apply shape optimisation to the more complex 3D models, something not available to the more rigid sizing models of the other levels.

The ability to undertake more complex analyses comes with it the drawback of an increased computational cost in both simulation time and memory resources dedicated to solving the model equations of the device or system. This can make automated design synthesis of MEMS that use modelling tools from the physical level harder as less time can be spent in developing solutions. Table 2.3 provides a summary of some of the major developments in physical level modelling and analysis, also included are some tools which span all levels of design.

2.1.4 Process level modelling and analysis

The process level looks towards the creation of appropriate mask layouts and / or process information needed for the batch processing techniques generally employed for MEMS device fabrication. Process level modelling and analysis tools look to simulate the processes and fabrication of MEMS often through the use of 3D numerical simulation of the chemistry and physics associated with wet etching and deposition found in anisotropic etching processes [47].

A number of modelling and simulation tools have been developed for this level allowing designers the ability to go from mask layout to solid 3D representations [92][93], and gauge the costs of fabrication within the design process [94]. A number of process modelling tools found at this level are shown in table 2.4.

Though the process level is of interest to MEMS designers, the bulk of automated design is associated with the system, device and physical levels and therefore the process level plays no further role in this thesis.

2.2 Conventional automated design synthesis and optimisation of MEMS

Though a number of MEMS simulation tools are available, these generally only allow for the modelling, visualization and analyses of device behaviour at varying levels of abstraction but do little with regards to optimisation. In order to allow parametric optimisation deriving several designs and running several simulations to determine which is the most optimal is not practical and foremost computationally expensive. The manual design of a MEMS device can be undertaken by an experienced designer however due to the multi faceted nature of a MEMS device experience in one field of design may not be sufficient as it may encompass several domains.

Table 2.4 Chronological review of developments in MEMS process level modelling and analysis

Contributors	Date	Methods	Applications	Description
Koppelman, G. K. [92][93]	1989	Process modeler, numerical	Gear and shaft	A process and fabrication simulator based upon IC fabrication techniques along with integrated sold modeler
Buser, R. A., et al. [95][96]	1991, 1992	Process modeler	Microresonators	ASEP, an integration into the CAEMEMS framework, geometric shapes can be outputted for FEA
Gilbert, J. R., et al. [97]	1992	Process modeler, numerical	Comb drive	Process modeler integrated into MEMCAD framework, allows CIF formats for structure construction along with electrostatic and mechanical analysis
Osterberg. P. M., and Senturia. S. D. [98]	1995	Process modeler, numerical	MEMS gyroscope	MEMBulider, a process modeler tool that allows the construction, visualization and analysis of solid models within MEMCAD
Hubbard, T. J., and Antonsson, E. K. [99]	1996	Process modeler	Three dimensional hole	Hybridizing geometrical and cellular approaches to process modeling, coined 'Directed Segs' designed to overcome previous work [65][66][67]
He, Y., et al. [100]	1996	Process modeler, numerical	Comb Drive	IntelliCAD, Process modeler and solid modeler tool
Zhang, C., et al [101]	2006	Process modeler, numerical	Numerous	An add on to the popular SolidWorks [69] CAD tool to allow mask layout and process information to be created from a solid 3D model

The aim of optimisation is to find the best possible decision from a set of possible choices without the needed to check every possibility within the search space. This can be broken down into three basic components: the objective function for the problem that we want to minimize or maximize a set of unknowns or variables which affect the value of the objective function; and finally a set of constraints that set conditions on the values certain variables are able to take.

Within traditional MEMS design optimisation the usual approach can be summarized into several steps. A model or representation of the device a designer wishes to optimise is constructed and parameterised.

A cost function used to evaluate the device design performance is formulated, and any constraints associated with the design are noted. Afterwards a mathematical model or formulation is built using the dependence of the cost function to the parameters of the device. Next by finding the parameters that optimise the device design with respect to any imposed constraints and evaluated against the chosen cost function becomes the final step [102].

Typical approaches generally loop through all possible design parameters until the optimum set of design parameters are found. Limitations with this are the computational cost is expensive and with each additional design parameter this increases. Also with a parameterisation scheme such as this it may not be possible to iterate along a continuous function of possible design parameter choices as they may lay outside the scope of the algorithm [102]. There may also be some wasted search effort, as not every design parameter will play as large a role as others in the overall cost function.

The conventional algorithm of choice for automated design synthesis and optimisation of MEMS is often some form of local search gradient based algorithm. Such algorithms however have a number of frailties, especially with MEMS as the design search space is often highly multi-modal, discontinuous and parameterized models often employ both discrete and varied design parameters.

Some of the major developments and examples of conventional automated design of MEMS are presented in table 2.5.

Table 2.5 Chronological review of developments in conventional MEMS design optimisation

Contributors	Date	Level	Methods	Applications
Haronain, D. [103]	1995	Device	Gradient search	Sensors, microactuator
Iyer, S., et al. [104][105]	1997	Device	Multi-start gradient search	Microresonator
Krassow, H., et al. [106]	1998	System, Physical	Gradient search	Pressure sensor
Ye, S., et al. [107]	1998	Physical	Gradient search	Comb drive, finger shapes
Mukherjee, T., et al. [108]	1999	Device	Multi-start gradient search	Microaccelerometer
Gibson, D., et al. [109]	1999	Device, Process	Incremental search	Micromirror
Parkinson, M. B., et al. [110]	2000	Device	Gradient search, simulated annealing	Bistable micromechanism
Sedivec, P. J. [111]	2000	Device	Gradient search	Crab-leg resonator, microresonator
Han, J. S., and Kwak, B. M. [112]	2001	Physical	Gradient search	Microgyroscope
Brenner, M. P., et al. [102]	2002	Device	Gradient search	Relay switch
Peano, F., and Tambosso, T. [113]	2005	Device	Numerical ODE method	Microconverter

2.2.1 Limitations of conventional design of MEMS

As has been highlighted throughout the development of computer-aided tools for the synthesis of MEMS the main focus has been upon the analysis of design specifications at varying levels of granularity or model level. Though crucial in the development of novel devices by allowing the

visualization and analysis of new designs it does little in driving forward an automated process of design optimisation. To date most optimisation is manually driven through a slow process of design, model evaluation and then re-evaluation, normally encompassing a host of software tools

Design synthesis methodologies, especially those that exploit all levels of design are still in their infancy [16] and the ability for designers to shift among the different levels of design abstractions without difficulty is important for improving the design process [53]. Often the conventional approach to MEMS design for a device or component relies purely on a MEMS designers experience and prior knowledge of similar devices [114] a view shared by a number of researchers. The design process is often performed in a 'trial and error' fashion which Zha and Du [115] describe as requiring several iterations before the performance requirements of the device are fulfilled, with Calis and Desmulliez [116] and Fan *et al.* [53] describing such an approach as highly costly, and very ineffective for commercial MEMS technology. There are also limitations in processing and fabrication of MEMS with current available simulation tools unable to predict the effect of complex external or internal environmental parameters or process variation [116].

There have been moves to incorporate automated optimization into the design process of MEMS with the integration of local search based algorithms however its penetration into the field of MEMS is still small. Though automated local search based optimization methods are often better at finding optimal solutions than through manual design iteration performed by a designer there are a number of drawbacks, often tied to the field of MEMS itself. The systems and devices with MEMS are often multidisciplinary and their behaviour highly non linear which can give rise to a design search space that is multi-modal, constrained and discontinuous. Local search algorithms are named as such because they often focus upon local regions of the design search space, and using gradient information transverse the space to find ever more optimal solutions. This is problematic as multi-modal search spaces often have many local optima scattered throughout the feasible design space open to the optimizer which can lead to such gradient based algorithms to become trapped and unable to transverse to what is the true global optimal solution design space. Also as most parameterised models within MEMS may contain both discrete and continuous design variables there are discontinuous regions for which the algorithm can fail to overcome. There has however been some progress in trying to overcome these particular problems through the use of a multi-start grid strategy which provides several starting points for the gradient based algorithm to act upon when optimising solutions to the design problem [30][105][108].

MEMS design optimisation is also often multi-objective with designers looking to improve performance or match synthesis targets while also trying to cut fabrication costs. With the exception of certain multi-start grid algorithms most local search gradient based algorithms focus on a single solution and most multi-objective problems have to be formulated as a single objective. Also given the deterministic nature of gradient based algorithms, repeated optimization attempts will lead to the same solution unless some form of multi-start grid strategy is used.

Unsurprisingly alternative approaches to design optimisation have been incorporated into the MEMS design process to overcome some of the frailties of the more conventional local search methods. These often stem from the more global and stochastic algorithms available within soft computing, and focus on a number of issues related to the MEMS design process, from optimisation to conceptual design.

2.3 Unconventional automated design synthesis and optimisation of MEMS

In order to improve the MEMS design process, in particular with regard to automated design synthesis and optimisation a number of researchers have put forward or applied unconventional techniques as a means to tackle previous problems of conventional optimisation algorithms or simply improve and push forward the optimisation of MEMS.

The design of MEMS devices can require very specific knowledge of the domains and applications that encompass the device which is to be designed which often falls upon the designer themselves. This can be made even harder when once designed a whole new set of skills are often required to produce the required process and fabrication information needed to build the device itself. In order to fill such a gap Gogoi *et al.* [117] developed an approach utilizing a direct acyclical graph (DAG) topological sorting algorithm to derive the optimal process sequence for fabrication of a 2D representation of a MEMS device.

Fabrication of MEMS also has a separate effect on how to go about optimisation of MEMS through the introduction of processing and fabrication errors. Even after a designer has produced an optimal MEMS design there is still the need to bring it into reality through fabrication and in doing so the processes of etching and deposition can introduce tiny but significant derivations or errors to the overall sizing and shape of the device. This is what motivated Shavezipur *et al.* [118] to use a probabilistic approach to the design optimisation of a parallel-plate capacitor and in doing so were able to reduce the effect of process variation on the yield of the optimal design.

Conventional gradient based methods of optimisation generally focus upon the sizing or shape of a particular device or component at the later stages of design often where a finalised topology or structure has been laid down. Conceptual design optimisation in MEMS is an interesting area of research which looks to the early stages of design with the goal of automatically creating novel topologies or structures to particular problems. Work by Campbell [119][120] looked to integrate an agent based automated design approach to the MEMS design process, in this instance the creation of novel ADXL accelerometer [121] designs. Various agents could employ a number of strategies to build and connect various components into new and unforeseen accelerometer designs.

Stochastic approaches however are the biggest set of unconventional optimisation strategies to be employed to MEMS design, with simulated annealing employed on a number of occasions [4][31][43]. The approach looks to simulate the effects of annealing in metal by having a stochastic process of selection employed to a parameterised model and over time reducing or ‘cooling’ the level of random selection probability. Therefore a level of global search can occur

early in the design optimisation process as there is a higher probability of a random solution getting chosen which lies further away in the search space then the current best design. Later on the selection probability is reduced so that solutions of higher fitness but also nearer to the current best solution are chosen, akin to hill climbing or local search. Such an approach is particularly useful for discrete and multi-modal design problems like those found in MEMS design.

A number of examples of unconventional automated design optimisation of MEMS are presented in table 2.6.

Table 2.6 Chronological review of developments in unconventional MEMS design optimisation

Contributors	Date	Level	Methods	Applications
Gogoi, B., et al. [117]	1994	Device	Directed acyclical graph	Planar devices, micromotor
Campbell, M. L. [119][120]	2000, 2001	Device	Agent based	Microaccelerometer
Kamalian, R., et al. [31][43]	2002, 2004	Device	Simulated Annealing	Meandering resonator, electrostatic actuator
Ongkodjodjo, A., and Tay, F. E. H., et al. [3]	2002	Device	Simulated Annealing	Microgyroscope
Shavezipur, M., et al. [118]	2008	Device	Probability distributions	Parallel plate capacitor

Though the stochastic method of simulated annealing has been used successfully in the past for MEMS design optimisation, the majority of the unconventional and stochastic techniques used in MEMS design come from the field of evolutionary computation, particularly single and multi-objective genetic algorithms.

2.4 Evolutionary computation for MEMS design synthesis and optimisation

MEMS design synthesis and optimization as an area of interest covers a wide range of topics applicable to current natural computing methodologies such as evolutionary computation. These include but are not restricted to: conceptual design [120], component based design [42], classical shape [107], sizing [122,123] and topological [43,124] design optimisation, multi-disciplinary [40], multi-objective [31,33], and multi-level design optimisation [50,53], hybrid genetic algorithms [124], robust design [123], interactive evolutionary algorithms [43], and case-based reasoning [125]. The bulk of work performed on MEMS through natural computing has focused on, predominantly, multi-objective evolutionary algorithms [31,33]. Early applications of genetic algorithms in the MEMS field were applied to the process level of design. Multi-objective genetic algorithms (MOGA) were later used for the evolution of simple mask shapes [41,126].

Advantages to using evolutionary algorithms over the more traditional gradient based optimisation algorithms, are that they are less prone to local optima entrapment, they can be:

- Hybridized with other methods [127]
- Self-optimisation [128]
- Parallelisation [129]

- Discrete variables and discontinuous search spaces.

An approach to overcome the limitations of traditional optimisation techniques, work undertaken by Zhou [32] applied MOGA at the device level to a number of examples (in structural design optimisation) using the Sugar modelling platform [66]. The success of MOGA over traditional and simulated annealing methods is apparent [31] with its ability to provide a set of Pareto optimal solutions and its ability to overcome some of the common pathologies of traditional gradient based search, such as a multi-modal and discontinuous search space. A number of limitations from this work highlighted a drawback, common in computer-aided design, of not being able to capture the complete behaviour of the device as accurately as possible, due to the limitations of the Sugar simulator. This can lead to designs which theoretically perform optimally, but upon further inspection by a MEMS designer, contain infeasible or constrained elements which were overlooked [122].

An example can be seen in [130] where evolved designs, here a meandering spring, contained topologies which were infeasible due to the high level of stress placed on them. This was discovered during verification analysis using ANSYS [89].

To improve upon this an interactive evolutionary algorithm approach, developed by Kamalian [43], was used in the design optimisation of a micro resonator. Here MEMS designers can interact online with the optimisation process, where a selection of current optimal solutions is periodically presented. Designers can then provide a ranking value for each design to qualitatively eliminate bad designs based upon designer knowledge. Kamalian also explored the use of constraints in aiding design through an exploration of a number of topological restrictions to a meandering resonator. Results showed that these restrictions aided the convergence to optimal designs by reducing the computational expense, in comparison to a non-constrained approach [33].

The standard MOGA is a global optimizer, which is often able to find a good approximation of the optimal design(s) but can struggle to reach them due to the local search operators used within them. Furthering work done by Kamalian [43], Zhang [124] incorporated a local optimizer, in this case a traditional gradient based method, with the more global MOGA optimizer to improve the final designs produced by the MOGA algorithm. The outcome highlighted that by allowing the MOGA to evolve good designs at both a topological and sizing level, and then later refining certain sizing design variables, superior results could be achieved to those accomplished by a simple non-hierarchical MOGA [42]. This approach was also extended into the use of interactive evolutionary computation through the creation of an interactive hybrid genetic algorithm [124]. Similar to past work [43] the main optimisation routine is driven by a MOGA, while periodically results are presented to a designer who can then initiate a local optimisation of their chosen designs [124].

Current work has somewhat followed the trend into hierarchical MEMS design optimisation, however rather than focusing on one level of modelling and analysis and utilizing a hierarchical search approach through global MOGA and local gradient based optimizers, multiple levels of modelling and analysis are used. Both [50] and [53] designed a MEMS bandpass filter utilizing, in a hierarchical approach, both system and device level modelling and analysis tools.

Fan [52,53] used a bond graph representation and a genetic programming approach to evolve a system level abstract MEMS bandpass filter. Though the system level representation allows for the function to be evaluated and evolved, the form of the device, in this instance the 2D layout of the individual folded flexure resonators, could not be evolved. The work demonstrated the eventual design synthesis of a folded flexure resonator to match target values; however the target values were not taken from the actual optimized system level filter design.

To overcome these limitations Farnsworth et al. [50] outlined an approach which linked both levels of design by converting system level electrical equivalent values to mechanical equivalent values using the equations proposed in [131]. Through the use of a MOEA, in this instance NSGAII [44], they were able to provide state of the art results with a significant reduction in the functional evaluations required to evolve a number of bandpass filter designs. These optimised electrical equivalent models were then converted to equivalent mechanical values for 'mass' and stiffness' and were then used as targets for the design synthesis of individual folded flexure resonators evolved on the Sugar platform.

Continuing work on folded flexure evolutionary design Fan [123] investigated the use of robust design algorithms for the application of MEMS design synthesis. MEMS fabrication is commonly associated with the deposition and etching practices of the integrated circuit community. This process can introduce errors which result in designs, or specifically sizing parameter values, to be inaccurate, [123]. Small changes in design can have large effects in device function and therefore it would be beneficial to be able to design robust device functionality prior to fabrication, to accommodate for manufacturing tolerances. The application of robust design methods can lead to solutions which are tolerant to varying sizing values in comparison to designs derived utilizing normal approaches in the design process [123].

The effect of small changes to sizing design variables is an important part in MEMS design optimisation as has been shown in [123]. The incorporation of local gradient based search algorithms into the design process allows for a much greater impact on local search than would otherwise be achieved with a standard MOEA routine. However incorporating local gradient based algorithms into hybrid MOEAs can be difficult and often require ad hoc constructions of the design problem for the local level optimiser. One similar approach highlighted by Fan et al. [133], as a future direction for improvement in MEMS design optimisation, are 'Memetic Algorithms' (MA) [134,135] and the local search heuristics in which they commonly use. A number of examples of evolutionary computational automated design optimisation of MEMS are presented in table 2.7.

The focus of the remaining chapter however is on evaluating the role multi-objective genetic algorithms can play in the design optimisation of MEMS. This includes the construction of a design optimisation platform which couples MEMS modelling and analysis tools with a number of state of the art multi-objective genetic algorithms for design synthesis and optimisation of MEMS.

Table 2.7 Chronological review of developments in evolutionary computational MEMS design optimisation

Contributors	Date	Level	Methods	Applications
Li. H., and Antonsson, E. K. [126]	1998	Process	GA	Simple mask shapes
Li. H., and Antonsson, E. K. [41]	1999	Process	MOGA	Simple mask shapes
Ma, L., and Antonsson, E. K. [136]	2001	Process	MOGA + noise	Simple mask shapes
Zhou, N., et al. [137]	2001	Device	MOGA	Meandering springs
Kamalian, R., et al. [31]	2002	Device	MOGA, GA	Meandering resonator
Zhou, N., et al. [32]	2002	Device	MOGA	Meandering resonator
Fan, Z., et al. [52]	2003	System	GP + bond graphs	Micromechanical bandpass filter
Kamalian, R., et al. [122]	2004	Device	MOGA + constraints	Meandering resonator
Kamalian, R., et al. [33]	2004	Device	Interactive MOGA	Meandering resonator
Kamalian, R. [43]	2004	Device	MOGA + object-oriented data structure	Meandering resonator, microaccelerometer, microgyroscope
Fan, Z., et al. [138]	2004	System, Device	GP + bond graphs + GA	Micromechanical bandpass filter
Kamalian, R., et al. [139]	2005	Device	MOGA	Microresonator
Fan, Z., et al. [123]	2005	Device	MOGA	Microresonator
Cobb, C., et al. [125]	2006	Device	MOGA + CBR	Microresonator
Zhang, Y. [124]	2006	Device	IHC	Microresonator, microaccelerometer
Zhang, Y., et al. [42]	2006	Device	MOGA + gradient search	Microresonator
Lohn, J. D., et al. [140]	2007	Device	GP + constraints	Meandering resonator
Hornby, G. S., et al. [141]	2008	Device	GP + noise	Microresonator
Farnsworth, M., et al. [50]	2010	System, Device	MOGA	Micromechanical bandpass filter

2.4.1 Multi-objective design synthesis and optimisation of MEMS

Traditional optimisation methods which utilise deterministic local gradient based search have often been the approach of choice when designers look to more automated approaches rather than hand driven optimisation. However because MEMS are inherently multi-disciplinary, with often complex nonlinear search spaces as a result of modelling analysis, there has been a shift to more stochastic automated optimisation techniques.

Stochastic methods such as those employed within the field of evolutionary computation have proven themselves successful on a number of MEMS design problems and in particular the application of single objective genetic algorithms and multi-objective genetic algorithms.

Genetic algorithms [142][143] are a class of population based global search algorithms from the field of evolutionary computation which look to mimic the processes of Darwinian evolution. The design problem is often encoded into a suitable representation and then a population of such solutions are iteratively evolved over a number of cycles until some stopping criteria are met as shown in figure 2.1. There are a number of operators present within the literature for selection, variation, often consisting of crossover and mutation, and finally replacement, a number of which are discussed in [144].

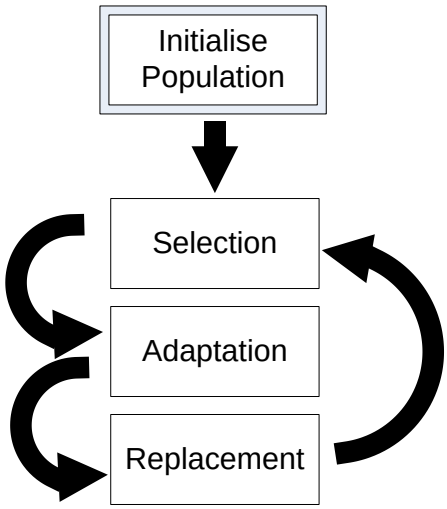


Figure 2.1 Evolutionary processes of standard genetic algorithms

Multi-objective design problems can be solved using the standard genetic algorithm through the combination of the objectives into a single weighted sum function. However it can be difficult to derive the correct weights for each objective and therefore it is often best to tackle the design problem with a multi-objective genetic algorithm.

Multi-objective genetic algorithms exploit the concept of Pareto optimality in order to partition the population of solutions into a number of ranked sets. This Pareto ranking of a population set works by utilising Pareto dominance to define a set of solutions which either dominate or are equal to all other solutions for each objective within the design problem. The first set to meet these criteria is given a rank and is defined as the Pareto optimal set. The process is repeated for the remaining solutions within the population until all ranks are filled as shown in figure 2.2. Where Pareto dominance can be used to define a set of optimal solutions within a multi-objective search space, it can also discourage the diversity of the population spread along each of the Pareto fronts. Therefore a number of diversity strategies such as crowding distance [45] have also been incorporated into recent state of the art multi-objective algorithms.

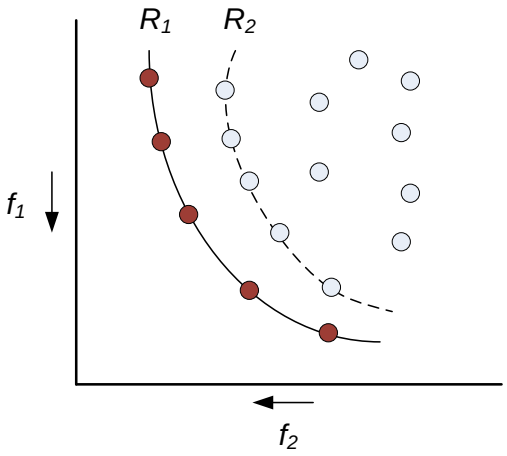


Figure 2.2 Pareto ranking within a set of solutions

In order to assess the performance of multi-objective genetic algorithms outlined a quantitative method suitable for multi-objective problems are required. This assessment should look to include two criteria when evaluating a final Pareto set:

1 The distance of the Pareto set and the individuals within it to the true Pareto front of the design problem.
2 The distribution of the individual solutions of the Pareto set along the front obtained

There are a number of metrics which measure these particular assessment characteristics within the literature [148][149][150][151], some require the use of the 'true' Pareto front for the particular problem, however in many engineering design problems this is not possible to know. The hyper volume or *S* metric [148][149] is an approach which utilizes a 'Nadir' point within the objective space and a supplied Pareto set to calculate the dominated subspace between both sets of data. Shown in figure 2.3 is an illustrative representation of the hyper volume indicator between two different Pareto sets, A and B. Using a provided reference point the hyper volume can be calculated for each of the Pareto sets provided and is relative to the total area of dominance between the Pareto set and the reference point used. The reference point must be dominated by all other points. It is clear from the figure that Pareto set A both dominates Pareto set B but as a result has a much larger subspace area of dominance. The hyper volume metric provides a quantitative value to the quality of the solutions found in a multi-objective problem.

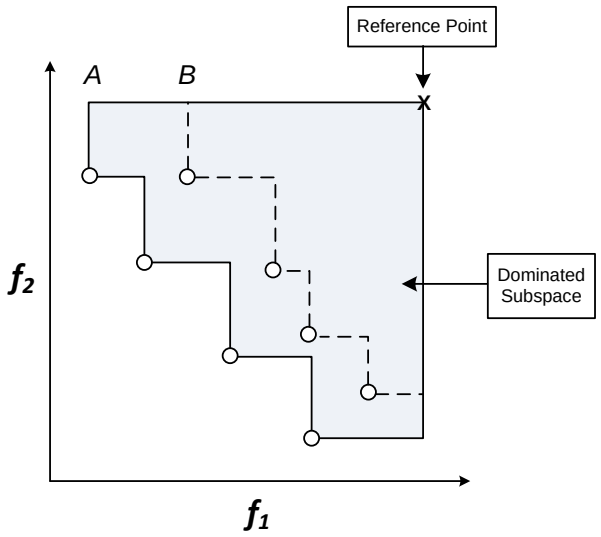


Figure 2.3 Hypervolume illustration for separate Pareto sets A and B

Once a final set of Pareto optimal solutions have been found there is the task of choosing the best solution applicable to the designer. This is made difficult simply because due to the Pareto nature all solutions are equally 'good' in relation to each other. In a 'a priori' approach a designer may favour one objective over others whether it is related to 'Cost' or 'Performance' and arbitrarily choose the best solution from this objective. In a 'a posteriori" approach a representative set of Pareto optimal solutions are selected and presented to the designer to choose from.

Though multi-objective genetic algorithms have been applied to the design optimisation of MEMS, the majority of this research has been derived from the work undertaken at the BEST

group within the University of California, Berkeley using a specifically designed and out of date MOGA incarnation.

To overcome this limitation appendix A covers a series of experiments covering five individual MEMS design optimisation problems used to evaluate a number of state of the art MOEAs in the literature. This provides an updating of the field of MEMS design optimisation through a further evaluation of the role MOEAs can undertake, along with a chance to construct the routines needed to link the optimisation algorithms with the various modelling and analysis tools present within the field of MEMS. Two algorithms are used during this early experimentation, MOGAII [44] and NSGAII [45], and an outline of their function is given below in algorithms one and two.

Algorithm 1: MOGAII Pseudo Code

1. Initialise population
    (a) Generate random population of size N and elite set $E = \theta$
2. Evaluate objective values
3. Assign rank based on Pareto dominance – 'Sort'
4. Generate offspring population
    (a) Combine both population and elite sets $P^{'} = P \cup E$
    (b) If the cardinality of P' is greater than the cardinality of P reduce P' removing randomly the exceeding points
    (c) Compute the evolution from P' to P'' applying MOGA operators:
        i. Randomly assign one operator (local tournament selection, directional crossover, one-point crossover or bit flip mutation) based upon probability of invocation
5. Evaluate objective values of population P''
6. Assign rank to P'' individuals based on Pareto dominance – 'Sort'
7. Copy all non-dominated designs of P'' to E - 'Sort'
8. Update E by removing duplicated or dominated designs
9. Resize the elite set E if it is bigger than the generation size N removing randomly the exceeding individuals
10. Return to step 2 considering P'' as the new P until termination

---

Algorithm 2: NSGAII Pseudo Code

1. Initialise population
    (a) Generate random population of size N and elite set $E = \theta$
2. Evaluate objective values
3. Assign rank based on Pareto dominance – 'Fast-Sort'
4. Generate offspring population
    (a) Create population P' using tournament selection and apply variation operators (Simulated binary crossover and polynomial mutation)
5. Evaluate objective values of population P'
6. Combine both population sets P and P' to give set size of $2N$ P''
7. Assign rank to P'' individuals based on Pareto dominance – 'Fast-Sort'
    (a) Fill new P set with non-dominated fronts until cardinality is reached from set P''
    (b) If cardinality of new set P is greater than the size N reduce P by computing the crowding distance of the last front set to be added and fill remaining slots using crowded-comparison operator
8. Return to step 4 until termination

2.5 Summary

A critical literature review was undertaken on the field of MEMS design synthesis and optimisation. This encompassed a review of the various modelling and analysis methods open to a designer and the advantages and limitations each approach brings. This was followed with an overview of the main traditional and non-traditional design optimisation methods used within the industry, with a particular focus upon automated stochastic methods from the field of evolutionary computation. From this a series of experiments were used to evaluate and update the role current state of the art MOEAs can play in MEMS design synthesis and optimisation.

Results showed successful application of MOEAs to several case studies of increasing design optimisation complexity. I particular the final case studies for the design optimisation of a folded flexure resonator and ADXL150 accelerometer showed a marked improvement over the current state of the art designs within the literature [4][42][53]. Having then established a routine for using the current state of the art multi-objective evolutionary algorithms towards MEMS design and evaluating and validating their performance for use within this thesis, the next step looks to explore the role multi-level and multidisciplinary optimisation strategies can play in MEMS design.

3

Literature Review on Multi-Level and Multidisciplinary Design Optimisation

This chapter provides an overview of the fields of multi-level and multidisciplinary design optimisation. The first section begins with a review of the field of multi-level design optimisation with special focus placed upon the methods of multi-level evaluation, parameterization and search. The final section concerns itself with a review the field of multidisciplinary optimisation with particular focus on population-based multi-objective methods.

3.1 Multi-level design optimisation

The nature of a multi-level schema can be found in many different and often wide ranging fields such as more traditional areas of application in design engineering [152] and combinatorial optimisation [153], to more exotic regions of economics [154] and biochemistry [155].

The nomenclature used to describe applications of multi-level methods is also very broad, with terminology such as multi-scale [156], multi-level representations [157], [158], multi-level simulations [159], multi-level programming [160-163] or multi-stage [164] just some of the example variations of a similar thread of work. Therefore it is reasonable to suggest that that no clear and standard definition of what constitutes a multi-level method, particularly when applied to design optimisation, consequently exists.

One particular definition however casts multi-level optimisation as an instance of searching in a dynamic environment. In these environments either or the entire objective function, decision variables and search space for a given solution does not remain constant with time [35][165].

In the context of design optimisation, multi-level techniques look to take what at the outset is a complex or somewhat intractable problem and then depending on the methodology of the approach simplify or ease its eventual optimisation. The various approaches into which this can be achieved is best outlined in terms of a schism where design problems are either decomposed into smaller sub-problems such as often found in multidisciplinary optimisation (MDO), or those which focus on simpler hierarchical or distributed optimisation methodologies.

This broad definition can be used to describe a number of soft computing techniques that exhibit some form of these characteristics; these include but are not limited to, co-evolutionary learning, parallel and distributed algorithms, island model evolutionary algorithms, and hybrid evolutionary algorithms. These approaches have all been applied in some way to tackle complex design problems such as those found within engineering.

Co-evolutionary learning [166] for example undertakes a different approach to solution and population qualitative assignment, where a solutions quality or fitness is obtained with respect to other possible solutions in the search space [166]. The fitness function of solutions within the population is dynamic as it is dependent on other solutions within the population and can therefore change over generations. The approach coined 'cooperative co-evolution' by [167] emerged from the motivation to solve problems from engineering domains which contained a high level of modularity, and provided a appropriate framework for evolving solutions in the form of co-adapted subcomponents.

Parallel and distributed genetic algorithms are increasingly being used as a means for increased performance and distribution of cost when undertaking expensive CPU CAD/CAE driven design optimisation [168][169]. Island model evolutionary algorithms (IMEA's) look to contain a number of population sets within the optimisation routine compared with the traditional panmictic population approach of genetic algorithms [170]. They have shown to increase diversity throughout the whole design process and outperform traditional single population EAs. IMEAs have also shown promise in their use for separable optimisation problems, where individual subpopulations maintain a degree of independence allowing for search to be directed in different regions of the search space [171].

The more traditional view of multi-level optimisation however focuses on a set of hierarchical levels which in some way act towards improving the overall optimisation procedure. These levels though not restricted to, could be varying levels of model or problem representation, contain separate optimisation procedures, or possibly vary in accuracy and computational cost of analysis.

Broadly the traditional multi-level optimisation structure can be categorised into two distinct approaches when employing various levels within the design optimisation process. The more standard approach shown in figure 3.1a and in common with other distributed soft computing methods [166][167] involves the use of multiple subpopulations or demes, each often linked with a specific level [169][170][171]. Here demes represent a biological term for a local population. Another approach uses only a single population or solution in a temporal design process where each level is invoked upon a design or set of solutions in a stepwise fashion from start to finish as shown in figure 3.1b. Each level can either be populated with a single problem representation or a set of solutions depending on the chosen optimisation algorithm.

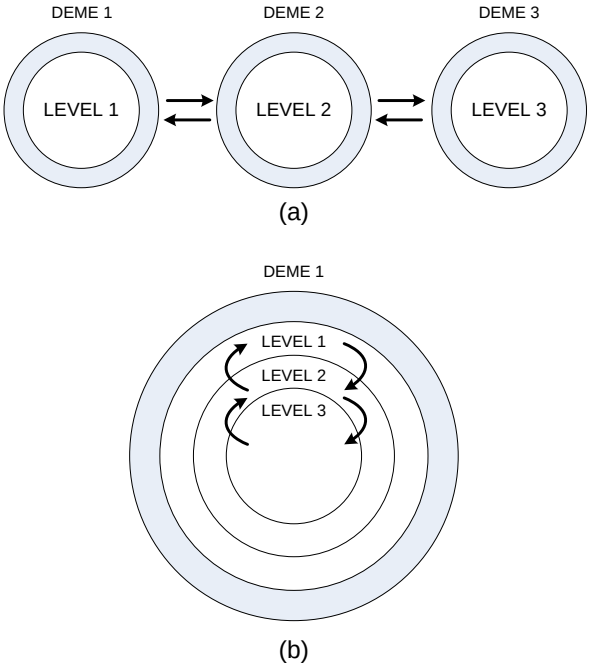


Figure 3.1 Multi-level approaches utilizing either an (a) 'island' approach or a (b) 'temporal' approach

Those multi-level structures which use a number of separate demes or subpopulations often require the transfer of information or solutions between each of these levels within the design optimisation routine. Therefore successful optimisation is also linked to migration of individuals / information between each subpopulation when compared to a closed 'partitioned' genetic algorithm which does not allow migration between islands [170]. Migration policies between the various levels within a population based multi-level algorithm vary depending on the construction and makeup. Multi population methods often employ a policy to transfer a set or dynamic number of solutions from one chosen subpopulation to another or from one master population to all others as shown in figure 3.2a. Single population strategies do not migrate solutions as in the traditional sense however policies exist on how many solutions and when a particular level is invoked as shown in figure 3.2b. This can include for example calling different analysis software or fitness evaluation such as in the use of in-exact pre evaluation (IPE), or in applying different search operators be they local or global over all or a truncated set of solutions.

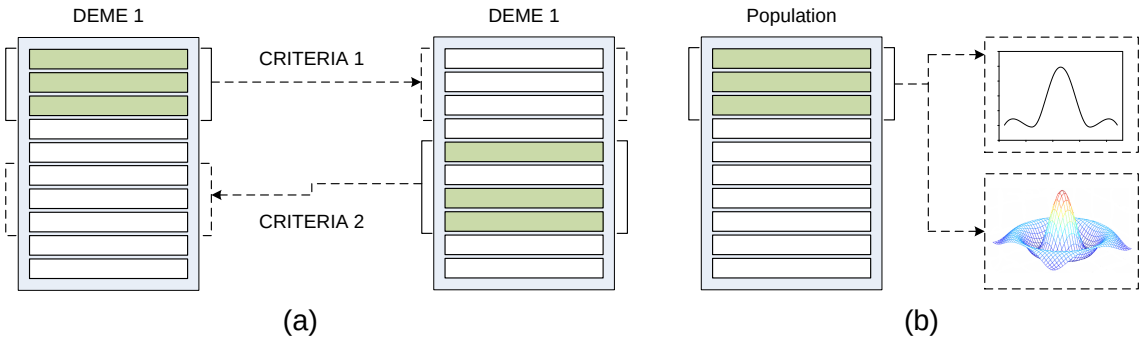


Figure 3.2 Migration policies for multi-level design involving (a) population transfer between demes / levels or (b) invocation of different levels within a single population.

This particular set of methodologies more closely mirrors the successful application of hierarchical design automation and optimisation found in electrical VLSI synthesis. This relates to the nature of VLSI design which contains many levels of modeling and analysis abstraction, similar to that found in MEMS design and where multi-scale algorithms can look to optimize at every level of abstraction, where each variable at any given coarser level represents a subset of variables at the adjacent finer level [156]. It is therefore a clear advantageous avenue for investigation when looking to improve MEMS design optimisation.

The area of multi-level design representation [157][158] masks the VLSI approach very well by defining the nature of most CAD and CAE tools in their level of abstraction in which they model and simulate objects. Often loosely formed as a hierarchy organized by the level of design abstraction, this can range from simple 'behavioural' level representations to more accurate 'functional' or 'structural' level representations. Each level has a natural relationship with those above and below it and there are often methods of communication or transfer from one representation to the other. These partitions often provide both benefits and tradeoffs, with high level 'behavioural' models often having lower accuracy in functional analysis but a significant reduction in computational cost, while low level design representations having high accuracy in terms of analysis but high evaluation cost [159].

One of the earliest forms of multilevel optimisation was undertaken by Dunham et al. [172] in which they worked with a two level problem where optimisation occurred at a lower level approximate model and then later refined at a higher level using a more accurate and computationally expensive model. Some work on injection island genetic algorithms [173], [174], [175] looked to emulate the process set out by Dunham by containing a majority of islands which contained computationally cheap but low accuracy evaluation, and a relative minority of islands which provided high cost and accurate evaluation and that were fed individuals from the lower cost islands.

Since then a number of multi-level design synthesis and optimisation has occurred the majority of which have been focused on macro level design problems in particular those associated with large scale systems or engineering structures. The aerospace industry features heavily, with multi-level approaches applied to the design of aircraft frame structures [52][176] that incorporate multiple levels of optimisation acting on whole and decomposed structural components to multi-level approaches which looked to overcome costly analysis in the design of aerospace engine systems [177][178]. A similar approach which looked to combine both multi-level evaluation and parameterization schema was also undertaken by [179] in the design optimisation of a space structure with the goal using approximate modelling to reduce computational cost of optimisation.

The bulk of recent multi-level optimisation work has focused on utilizing the techniques of distributed genetic algorithms (DGA) and a subset hierarchical distributed genetic algorithms (hDGA) [168][34][180] to solve a series of design problems. Found within the field of evolutionary computation these techniques contain similar characteristics to multi-level design optimisation, in particular the distributed nature of the algorithm through the use of several populations each of which is processed by a genetic algorithm independently from others. An additional 'migration' operator is enforced to allow chosen population members to be

exchanged between each of the subpopulations in effect hoping to increase genetic diversity and halt any unnecessary convergence of the system into local optima. Each distributed deme or population can be seen as its own separate level and therefore in a multi-level fashion possibly contain different optimizers, simulation tools, model representations or decision variables.

Table 3.1 Chronological review of developments in multi-level methods

Contributors	Date	Multi-level Methods	Applications	Description
Hill, D., and vanCleemput, W. [159]	1979	Evaluation	Numerous	SABLE, a multi-level structural and behavior linking environment
Nestor, J. A., and Thomas, D. E. [157]	1982	Parameterization + Evaluation	Digital design	Multi-level representation to aid system level designers
Ding, Y. [152]	1988	Decomposition	Numerous	Decomposition of complex problem into a multi-level hierarchy of simpler problems
Ding, Y., and Esping, B. J. D. [176]	1991	Decomposition + Evaluation + Search	Numerous	Improved [152] by incorporating multiple levels of evaluation / search
El-Beltagy, M. A., and Keane, A. J. [165]	1999	Evaluation	Bump problem	Temporal multi-level single population optimisation
Nair, P. B., and Keane, A. J. [179]	1999	Evaluation + Parameterization	Space structures	Co-evolutionary genetic algorithm structural design
Leary, S., et al. [177]	2001	Evaluation	Suspended beam	Coarse and fine modeling methods explored
Karakasis, M., and Giannakoglou, K. [34]	2003	Evaluation	Aerodynamic shapes	Surrogate evaluation models incorporated into hDGA routine
Maksimovic, S., and Zeljkovic, V. [178]	2004	Evaluation	Aircraft nose wheel	System and component level optimisation
Walshaw, C. [153]	2004	Refinement + Search	Combinatorial	Refinement and coarsening of multiple graph partitioning levels
Yanhong, Z., et al. [155]	2004	Decomposition + Parameterization	Genetic	Gene structure prediction using a hierarchy of decomposed units
Chan, T. F. et al. [156]	2006	Evaluation + Parameterization	VLSI physical design	Multi-scale application to VLSI design partitioning, placement and routing
Tiwari, A., et al. [154]	2008	Parameterization	Four stage rolling problem	Multi-stage optimisation of real-life rolling system design problem
Faisca, et al. [183]	2009	Decomposition + Parameterization	Numerous	Multi-level decentralized optimisation

This is exactly what was done in later work, with an exploration into the benefits of several multi-level strategies using distributed genetic algorithms towards design optimisation. Here three main approaches were outlined, a '*multi-level evaluation*' approach which broke up each sub population into separate levels of model evaluation, a '*multi-level parameterization*' approach which partitioned the problem into separate levels of model parameterization and finally '*multi-level search*' which utilized separate levels of search algorithms be they global or local optimizers [34][181][46].

To take it to its eventual conclusion an approach outlined in [182] combined all three strategies into one framework to solve a series of optimisation problems. The use of a single strategy showed over 50% marked reduction in computational cost over the standard evolutionary

algorithm approach and the final combined evaluation, parameterization and search multi level framework showed a 95% reduction in computational cost. Cost relates to the total computational expenditure in functional evaluations / analysis calls to find an optimal solution, relative to other strategies. Therefore a 50% reduction in computational cost involves finding a similar or superior solution to previous strategies using 50% less functional evaluations / analysis calls or the equivalent cpu cost in analysing solutions, often ranked in time required to analyse.

The next set of sections looks at each of these multi-level strategies in more detail and highlights more examples of their application in design optimisation in the literature.

3.1.1 Multi-level evaluation

Multi-level evaluation strategies generally focus on the use of a hierarchy of levels each of which contains a separate method of evaluation, most commonly ranging from a low fidelity but low cost evaluation tool to a high fidelity and high cost evaluation tool. Depending on the overall framework or methodology chosen the optimisation procedure may evaluate solutions at each of these different levels temporally, or for example in a distributed framework, separate levels may optimize solutions using local evaluation tools and then exchange the solutions to lower fidelity or higher fidelity levels as shown in figure 3.3. Different fidelity analysis in this instance may involve using a model which lacks the available disciplinary analysis found at the higher level, models which only perform partial analysis at a lower cost, or models which contain a lower granularity, for example partial meshing of a FEA model.

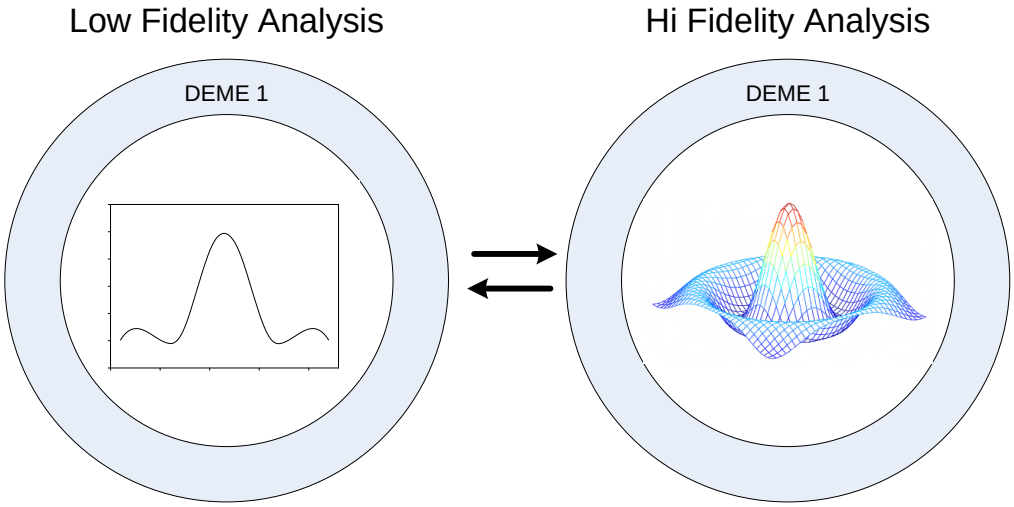


Figure 3.3 Multi-level evaluation: consisting of multiple levels of analysis software

This approach of dynamically interacting with various levels of evaluation accuracy stemmed from early work by [184]. Here a genetic algorithm approach was used to answer the question ‘Given a fixed amount of computation, is it better to devote substantial effort to seeking highly accurate evaluations or to obtain quick, rough evaluations and run the GA for many more generations?’. The outcome of this research showed that using the less accurate but computationally less expensive model over an increased number of generations was conducive to a better optimisation process then if more time was spent on more accurate evaluations over a smaller number of generations. This is naturally extended in a multi-level evaluation domain

which sees the majority of approaches look to utilize fast explorative search using low fidelity models and then later on perhaps finalizing designs using more accurate evaluation tools.

A multi-level approach towards the digital design process was outlined by [158], here the process looked to turn an idea into a physically realizable digital system following a top-down hierarchical methodology suitable to a multi-level design approach. Outlined was a series of abstract levels at which a designer can provide input or design can be represented, ranging from high level, computationally cheap representations (Behavioral, Functional, Logical) to lower level models which themselves more accurately represent the final product and therefore provide a higher computational cost (Gate, Circuit and Physical Layout). The authors looked to find common features in each representation which could be related to each other and define a hierarchical process for design optimisation. Similar top-down hierarchical frameworks have also been developed for VLSI design [185].

A number of other multi-level evaluation strategies employed within the literature are shown in table 3.2 covering a number of applications and techniques. The advantages of employing such a strategy to design often centre on the potential savings in computational cost when undergoing analysis of any new design. The authors in [186] use this as motivation when they describe that an ‘optimisation algorithm can often utilize relatively simple models to make search control decisions, and rely on complex models only when needed to verify optimality of a solution and satisfy constraints’. As a result by combining both models it is possible to derive designs as good as those found using only the costly model but at a significant reduction in computational cost.

The expensive cost of modelling and evaluation is exacerbated in two other areas of automated engineering design synthesis and optimisation, the application of population based optimisation algorithms to engineering design and the integration of expensive FEA/BEA software. Standard population based optimisation algorithms like those found within evolutionary computation often require a large number of functional analysis calls for the population of solutions used through the optimisation routine. However there is still a necessity of employing such algorithms for design optimisation due to their increased performance over more traditional methods. The increase in the number of functional evaluations is often made that much worse as designers often employ and require the use of computationally expensive FEA/BEA analysis.

A number of works have looked to directly tackle these issues through the use of multi-level strategies. Whitney et al. [169] proposed an EA method to the shape optimisation of aerofoils that looked to spread the search over three levels of increasing accuracy. This was in order to overcome the drawback of traditional EAs which exhibited high cost through often hundreds of expensive computational fluid dynamic (CFD) calculations. Results showed a marked decrease in the number of evaluations and CPU cost compared with the more traditional EA approach [169]. The reduction in analysis cost in both memory resources and time have also been addressed as it links heavily with the strategy outlined. In order to build a hierarchy of low and high cost models specific techniques such as response surface modelling (RSM) [187] or metamodels [34] are often needed. These low cost models allow some level of functional evaluation of solutions evolved by the optimizer to occur while reducing the computational cost to do so.

Table 3.2 Chronological Review of Developments in Multi-level Evaluation Methods

Contributors	Date	Applications	Description
Thomas, D. E., and Nestor, J. A. [158]	1983	Digital design	Extension of previous work [157] on multi-level representation
Philipson, L. [185]	1990	VLSI design	Multi-level design process based upon abstract functional models
Ellman, T. et al. [186]	1997	Yacht design	Multi-model strategy
El-Beltagy, M. A., and Keane, A. [35]	1999	Satellite boom	Generative topographic mapping (GTM) used to aid multi-level evolutionary search
El-Beltagy, M. A., and Keane, A. [165]	1999	Bump problem	Temporal multi-level evaluation using various low / hi cost analysis strategies
Whitney, E. J. [169]	2002	Transonic nozzle	Hierarchical GA using three levels accuracy, shows a 30% increase in speed over traditional GA
Karakasis, M., and Giannakoglou, K. [34]	2003	Aerodynamic shapes	Surrogate models used for pre-evaluation to reduce computational cost
Thiyagarajan, N., and Grandhi, R. V. [187]	2005	3D Steering Link	Response surface modeling for multi-level shape optimisation, 50% performance increase in optimal design
Kampolis, I. et al. [181][46]	2007, 2008	Transonic compressor	Meta-assisted EA combined with low and high cost analysis in hierarchical framework
Kampolis, I. C., and Giannakoglou K. C. [188]	2009	Numerous	Distributed hierarchical EA coupled with problem specific and local metamodels
Kampolis, I.C., and Giannakoglou, K. C. [182]	2010	Ackley function	A reduction of 50% CPU cost using multi-level evaluation strategy

3.1.2 Multi-level parameterization

Multi-level parameterization focuses upon two general concepts, the breakdown of a complex or intractable problem into a hierarchy of design / modeling levels where either there exist problem solutions with varying levels of decision variables or constraints present or each level focuses upon a different level of optimisation representation, be it topological, sizing or shape optimisation.

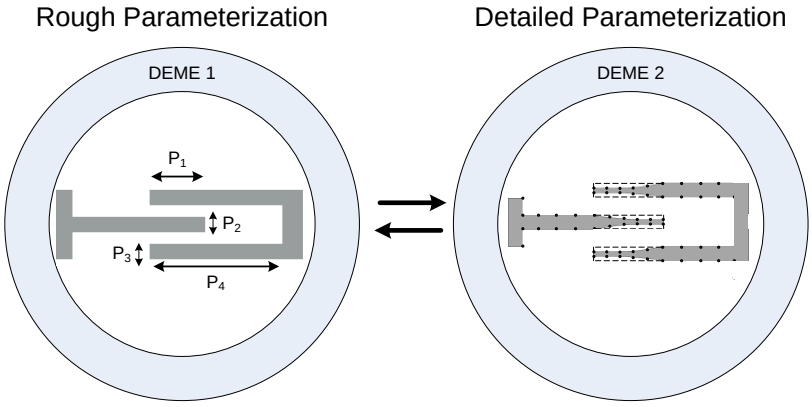


Figure 3.4 Multi-level parameterization: consisting of multiple levels of model granularity

The reduction in the dimensionality of the problem search space is one of the key motivators outlined in [182] for using such a strategy. Here a combined multi-level evaluation and parameterization strategy for solving a modified Ackley function was able to reduce the cost of optimisation by 75%, with the low level in a reduced dimension search space due to a lower number of design variables. A similar thread of work is also found in [181] and [46] where aerodynamic shapes are parameterized using Bêzier curves with either 7 (low level) or 17 (high level) points in a hierarchical evolutionary process that was able to outperform a similar single level EA.

Alternating between shape, topological and sizing optimisation are an alternative form of multi-level parameterization strategy employed in recent work by [189] with topological search and then shape optimisation used in a sequential optimisation procedure for optimal design of an electromagnet. Shape optimisation also features heavily with multi-level strategies significantly speeding up convergence to optimal designs in a number of design problems [36][38]. An approach outline by Duvigneau [37] looked towards implementing a multi-level approach towards the design optimisation of aerodynamic design. Here a set of differing shape representation levels, ranging from a course-grained level parameterization including a small number of design variables to a fine grained level parameterization which implies a large number of design variables were used in the optimisation procedure [37]. The overall approach can be seen as a journey through several embedded design search spaces of varying dimension and complexity. The approach outlined utilized a particle swarm optimisation strategy, with motivation being that traditional strategies such as multi-grid theory for Partial Differential Equations (PDEs) solving are dedicated to descent methods, which can easily be caught up in local optima [37]. The proposed approach yielded significant computational cost reduction. The approach also looked to yield a balance between the global searches in the space of higher dimension and use of prior searches within subspaces of lower dimension [37]. On a similar thread a global and local search method was developed by [190] and consisted of a simulated annealing algorithm modified for multi-level parameterization for the design optimisation of a series of truss bar test cases. The strategy proved successful over a number of state of the art algorithms with a significant reduction also in computational cost. The algorithm works by optimizing all design variables at once on one global level in order to meet some objective and then the best designs are optimized locally by looking to perturb individual design variables one by one.

There are a number of obstacles to overcome when often employing a multi-level parameterization strategy, in particular how to handle transformations between the various levels of modelling granularity. Transformation algorithms may be required [181] to change models migrating from low to high parameterization levels, while such changes can also affect the design search space the optimizers are acting upon which may also alter performance [191].

A list of further multi-level parameterisation strategies employed within the literature are shown in table 3.3 covering topological, shape and sizing design optimisation methods as well as a number of attempts to reduce the complexity of the design search space during optimisation.

Table 3.3 Chronological Review of Developments in Multi-level Parameterization Methods

Contributors	Date	Applications	Description
Li, Q. S. et al. [191]	2001	Steel frames	Multi-objective and multi-level optimisation of decomposed steel frames
Desideri, J-A., and Janka, A. [38]	2003	Airfoils	Outlined a full and adaptive multi-level optimum-shape algorithm (FAMOSA)
Desideri, J-A., and Janka, A. [192]	2004	Supersonic Jet	Multi-scale shape optimisation shows faster convergence then single level
Walshaw, C. [153]	2004	Combinatorial problems	Multi-level refinement method applied to graph partitioning and graph colouring
Soper, A. J., [193]	2004	Combinatorial problems	Extended [Walshaw] to use stochastic evolutionary algorithms
Abou El Majd, B. et al. [36]	2006	Aerodynamic wing shapes	A number of strategies 'parameterization adaption etc' used for multi-level design
Martinelli, M. And Beux, F. [38]	2006	2D nozzle	Improvement to gradient based optimisation through control of parameters
Duvigneau, R. [37]	2007	Aerodynamic wing shapes	Hierarchical shape parameterization strategies for aerodynamic design using particle swarm optimisation
Lukas, D., and Chalmoviansky, P. [189]	2007	Electromagnet	Combines topological and shape optimisation in a hierarchical optimisation
Kampolis, I. et al. [181][46]	2007, 2008	Compressor cascade	Blade airfoil design over two levels of parameterization, high level shows global search, low level reaches optimum rapidly
Korosec, P., and Silc, J. [194]	2008	Electric motor problem	Employed same multi-level refinement as [153] while using ant-sitgmergy algorithm, more successful than traditional methods
Lamberti, L. [190]	2008	Truss bar	Multi-level strategy involving perturbation of single or all design parameters
Liakopoulos, P. I. K., et al. [195]	2008	Compressor cascade, 3D elbow duct	Grid enabled hierarchical l EA shape optimisation
Kampolis, I.C., and Giannakoglou, K. C. [182]	2010	Ackley function, airfoil	A reduction of 75% CPU cost using a multi-level parameterization + evaluation strategy

3.1.3 Multi-level search

The hybridization of both global search algorithms and local search algorithms is perhaps not a new concept, and such hybrid algorithms have found common application in the field of evolutionary computation. Multi-level search as a strategy looks to incorporate several types of optimisation procedure over the various hierarchical levels. This can range from a standard global to local optimisation approach as shown in figure 3.5, or perhaps to one which utilizes the

benefits some specialist algorithms bring for example genetic programming's topological search coupled with a standard MOGA.

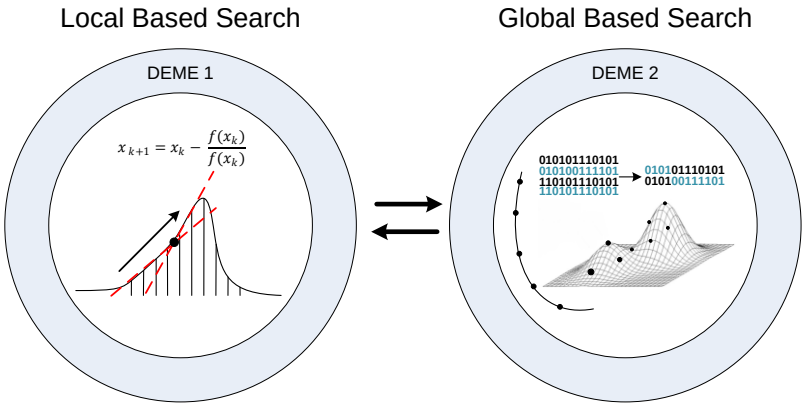


Figure 3.5 Multi-level search: consisting of multiple levels optimisation algorithms

Table 3.4 Chronological Review of Developments in Multi-level Search Methods

Contributors	Date	Applications	Description
Ding, Y., and Esping, B. J. D. [176]	1991	Numerous	Improved [152] by also integrating a separate method of moving asymptotes (MMA) [200] at the component levels.
Li, G., et al. [201]	1999	Steel frame	Decomposed problem with separate box complex and augmented lagrange multiplier optimisation methods
Giannakoglou, K., et al. [198]	2000	Aerodynamic wing shapes	Hybrid genetic algorithm and adjoint method for aerodynamic shape design
Lee, Y., and Lee, M., [199]	2002	Shape-based block layout	Tabu search / simulated annealing used for local and genetic algorithm for global search
Conceicao Antonio, C. A. [202]	2002	Composite structures	Multi-level evolutionary search, with separate EAs with different parameters
Yui, K., et al. [197]	2004	Numerous	Combined simulated annealing and gradient-based method
Hansen, L. U., and Horst, P. [196]	2008	Aircraft fuselage	Topological + sizing optimisation
Kampolis, I. C., and Giannakoglou K. C. [46]	2008	Isolated airfoil	Two level strategy with MAEA used for global and SQP for local search optimisation
Kampolis, I.C., and Giannakoglou, K. C. [182]	2010	Ackley function, airfoil	Multi-level search combined with multi-level evaluation and parameterization provides 90% reduction in computational cost

A good example of merging specialist algorithms can be found in [196] where a combined multi-level search and parameterization method was used to optimize an aircraft fuselage structure. In this approach a top level Evolutionary Strategy algorithm drives topology search of the design, while a second level of optimisation based upon a deterministic gradient-based approach is used to drive sizing design parameters. The authors looked to break up the design problem into two domains, that of a sizing task and a layout / topology task. Each level also contains a separate set

of design variables akin to multi-level parameterization and as a result, the optimisation routine looks to firstly optimize the layout of the design which are then used by the second level and optimized separately as a sizing problem. Experimental comparison with a standard single level evolutionary strategy approach showed marked improvement in reducing weight for a simple framework boom, and work on a design optimisation of a fuselage structure proved successful.

The majority of multi-level search strategies look to couple the fast exploration of global and fast refinement of local search algorithms towards design optimisation, with the most recent examples found in [46][182]. Often local gradient based search algorithms are used when the more global search algorithms have stagnated by refining the best solution found so far [197][198], or during initialisation of the optimisation procedure by improving the initial generation to reduce the design space [199]. A list of further multi-level search methods within the literature is shown in table 3.4.

The preceding multi-level evaluation, parameterization and search strategies encompass the majority of what is the more traditional multi-level optimisation methodologies present within the literature. However there is also another class of algorithm which exhibits the characteristics of multi-level optimisation. The field of multidisciplinary optimisation is a class of coordination algorithms that look to decompose complex problems into a number of hierarchical or non-hierarchical subsystems in order to improve the process of optimisation. The following sections focus upon this class of algorithm.

3.2 Multidisciplinary optimisation

Complex large scale systems found in many engineering problems today can consist of many components and disciplines coordinating together to form some function or behaviour. In a real design engineering problem, each discipline typically represents a design team concerned with the design of one aspect or component of this complete system. This makes perfect sense as it allows many more people to work upon a particular problem while also allowing specialized designers to focus upon their respective disciplines [211]. There are however drawbacks, with the possibility of each discipline having to interact with others the chances of infeasible / non-viable designs occurring due to conflicts with other engineering teams and their separate disciplines is possible [212]. This is often solved with a post-optimisation trade-off where in order to solve such inconsistencies and obtain a feasible design; changes need to be made which often lead to a sub-optimal design [212]. Therefore there is a need to both optimise the individual disciplines and their constituent parts or components all the while maintaining some level of global design optimisation for the system as a whole.

MDO is one such class of algorithm which looks to coordinate these individual disciplines and components towards a system design that is optimal as a whole and satisfies all constraints, while maintaining some level of design autonomy [40]. This often involves the decomposition of the original design problem into a set of hierarchical coupled elements often based upon the analysis techniques which are used to analyze the physical or behavioural characteristics of the system, or the possible different physical scales, components within the system. As such the total structural performance of the whole system can be a combination of responses that are evaluated from each level within the hierarchy [213].

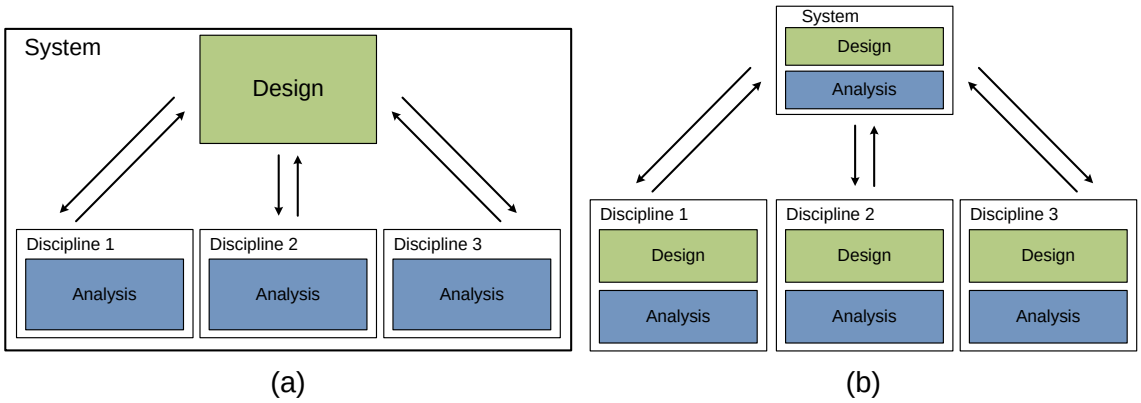


Figure 3.6 Disciplinary autonomy with (a) single-level analysis autonomy and (b) multi-level design autonomy

Once a hierarchy of decomposed elements is present their coordination and level of autonomy need to be assigned within the optimisation routine. The lowest level of control may be called 'analysis autonomy' where the role of each disciplinary group is limited to the selection and analysis of models [211]. The simplest examples are the single-level methods such as multi-disciplinary feasible (MDF), individual disciplinary feasible (IDF) or an all-at-once approach (AAO) [214] which generally focus upon a centralized decision making process at one level, where analysis can also be undertaken at each discipline or element as shown in figure 3.6a.

It is possible to improve these single level methods by utilizing multiple computers or grid systems for distributed analysis, and database management to give improved efficiency and maintainability. However the reliance on a single optimizer to act as a central decision maker and control all aspects of design for what is often a large scale and complex design problem is still a drawback [215]. The natural progression and next level of autonomy is the inclusion of both analysis models and optimisation algorithms in a distributed multi-level optimisation structure in what can be coined 'optimisation autonomy'. Here each level can contain its own set of analysis and optimisation routines and maintains some element of control over them as shown in figure 3.6b.

The coordination of a decomposed problem solution such that the overall global solution is found is a challenging task [216] however over the last thirty years a large body of work has been conducted towards this goal [217-221].

In the literature there are 6 main approaches to MDO which stand out from the rest; these are Optimisation by Linear Decomposition (OLD) [222], Collaborative Optimisation (CO) [223], Concurrent SubSpace Optimisation (CSSO) [224], Bi-Level Integrated System Synthesis (BLISS) [225], Analytical Target Cascading (ATC) [226] and the method of Quasi-separable Subsystem Decomposition (QSD) [227]. Each method differs in the way it coordinates the solution of a decoupled multilevel optimisation problem.

Newer methods include the inexact penalty decomposition method [228] and augmented Lagrangian coordination [229]. An overview of the current main approaches to MDO can be found in [213][230].

The approach coined optimisation by linear decomposition, first proposed by [231], this method was then extended into a general multi-level framework [232] and a number of successful demonstrations on various engineering design problems can been seen in [233][234][235][236].

The concurrent subspace optimisation (CSSO) method was first introduced in [224] and additional work on the handling of design constraints [237][238] and the sensitivities between systems [239][240] has also been undertaken. A successful implementation of the routine was shown in [241].

Collaborative optimisation [223] is a methodology for the decomposition of large-scale design problems into smaller design problems which can then be assigned and optimised by different groups of engineers. In real-world domains such as this it proves beneficial to decompose the problem, allowing each team to design separately speeds up the process due to the possible increase in people employed to the task and the ability to exploit the expertise of many specialized engineers [211]. In collaborative optimisation the problem is decomposed into generally a bi-level structure with individual subsystems tasked with satisfying local constraints utilizing local optimizers and analysers and permitted to vary local design variables [215]. collaborative optimisation closely resembles the formulation of OLD and has been applied to various large optimisation problems [242] and though successful, research has shown that it was unlikely to ever converge to an optimum [243].

A bi-level integrated system synthesis (BLISS) approach which looks to distribute part of the system level objective function to the subsystems was first introduced in [225] and extended later on for structural optimisation in [244] and has also seen successful implementation on a number of engineering design problems [221][245][246].

Table 3.5 Chronological review of developments in multidisciplinary optimisation

Contributors	Date	Description
Sobieszczanski-Sobieski, J., et al. [222]	1985	OLD, optimisation by linear decomposition
Sobieszczanski-Sobieski, J. [224]	1988	CSSO, concurrent subspace optimisation,
Cramer, E. J., et al. [214]	1994	IDF, individual discipline feasible method, each discipline is solved independently outside of system level
Cramer, E. J., et al. [214]	1994	MDF, multidisciplinary feasible method, each discipline is directly coupled in some way through input and output analysis, system level controls global / local design variables
Balling, R. J., and Sobieszczanski-Sobieski, J. [219]	1996	SAND, simultaneous analysis and design
Braun, R., et al. [223]	1996	CO, collaborative optimisation,
Sobieszczanski-Sobieski, J., et al. [225]	1998	BLISS, bi-level integrated system synthesis,
Kim H. M., et al. [226]	2003	ATC, analytical targeting cascade,
Haftka, R. T., and Watson, L. T. [227]	2005	QSD, quasi-separable subsystem decomposition,
DeMiguel, A. V., and Murray, W. [228]	2006	IPD, inexact penalty decomposition,
Tosserams, S., et al. [229]	2008	ALC, augmented lagrangian coordination,

Analytical target cascading (ATC), introduced by [226] looks to have the desired system response cascaded down the hierarchy from subsystem to subsystem and has been applied to various optimisation problems [247],[248],[249]. An approach called Quasi-separable Subsystem

Decomposition (QSD) looked to overcome problems of trapping in local minima of the search space as a result of decomposition and the effect coordination of the solution can have lead the creation of the QSD [227] algorithm. Its application to a number of structural optimisation problems are found in [250][251]. A list of the major MDO methods within the literature is shown in table 3.5.

3.2.1 Multi-objective multidisciplinary optimisation

The approaches outlined previously that include optimisation routines within all levels of the design optimisation framework generally utilize traditional gradient-based optimisation methods using a single solution only and focusing on a single objective. A number of authors have adapted these traditional methods to create multi-objective MDO formulations using a single weighted sum or aggregated objective [252][253]. However the majority of design engineering problems are highly complex with non-linear responses, discontinuous and multi-modal search spaces and contain both discrete and continuous decision variables. All these factor in a number of pathologies to the search efficiency of the more traditional optimizers, while single solution strategies only provide a single Pareto solution from each run for a designer to choose from.

Therefore looking to incorporate more robust population-based algorithms such as those found within the field of evolutionary computation that focus upon multi-objective design problems could be beneficial. An early example created by [254] featured an immune network system multi-objective genetic algorithm approach (MOGA-INS) for MDO designed to solve hierarchically decomposed multi-objective problems. Each decomposed unit or subsystem contained a MOGA which focused on a specific set of design variables held within the subsystem population representation. Limitations with this approach involved the need for each subsystem to contain the same objectives as all others and being limited to a hierarchical structure.

In order to overcome limitations from this previous work [254], the authors in [255] created a multi-objective multidisciplinary optimisation algorithm for hierarchically decomposed problems which allowed for differing objectives within each subsystem. This particular approach used quality metrics as a basis for objective function measurement for individual solutions at the system level.

Other multi-objective population based algorithms have been implemented within MDO over the years with varying degrees of implementation and success [256][257][258][259][260]. Perhaps the simplest instantiation of a multi-objective genetic algorithm MDO is presented in [261]. Here a hierarchical multi-level structure is used with a MOGA present in both system and subsystem levels. The system level holds the complete set of objectives and constraints outlined by the designer while each subsystem can consist of (additively) separable or unique objectives and constraints. The system level optimizer acts upon a set of local shared design variables while each subsystem contains their own set of specific disciplinary design variables. Each subsystems MOGA than begins an iterative optimisation routine for a set number of cycles before then passing on their evolved populations to be reconstituted with solution design variables from the other subsystems. This ‘grand pool’ of solutions is then evaluated using the system level objectives and constraints and then used to fill the next system level population using available replacement operators, in this instance a calculated entropy value. The algorithm was successfully tested against a well known speed reducer problem against a standard MOGA

showing superior performance. A list of multi-objective MDO algorithms found within the literature is shown in table 3.6.

Table 3.6 Chronological review of developments in multi-objective multidisciplinary optimisation

Contributors	Date	Applications	Description
Tapetta, R. V., and Renaud, J. E. [252]	1997	Numerous	Multi-objective collaborative MDO, system level contains a weighted sum of subsystem level objectives, subsystems aim to minimize interdisciplinary inconsistencies
Kurapati, A., and Azarm, S. [254]	2000	Speed reducer	MOGA-INS, immune network simulation method integrated with MOGA to give hierarchically decomposed MDO
Gunawan, S., et al. [255]	2003	Speed reducer, UAV payload	Hierarchically structured MOGA MDO, requires separable or additively separable objectives
Gunawan, S., et al. [261]	2004	Speed reducer	Hierarchical structured MOGA MDO, system level optimizer focuses upon shared design variables / objective while subsystem focus on local variables and objectives
Giassi, A., et al. [256]	2004	Roll stabilizer fin	MORDACE, a MOGA MDO that incorporates robust design with each discipline design solutions able to handle variation from shared data during a compromise at end of routine
McAllister, C. D., et al. [253]	2005	Race car design	Integrated linear physical programming with collaborative MDO
Aute, V., and Azarm, S. [257]	2006	Speed reducer, numerical test problem	Multi-objective collaborative MDO, system level optimizer focuses upon shared design variables / objective while subsystem focus on local variables and objectives
Rabeau, S., et al. [258]	2007	Speed reducer, dock design problem	COSMOS, collaborative optimisation strategy for multi-objective systems, optimizer focuses upon shared design variables / objective while subsystem focus on local variables and objectives
Huang, H-Y., and Wang, D-Y. [259]	2009	Container ship	Mixed weighted and multi-objective collaborative MDO utilizing multi-island genetic algorithms on all levels of design
Zadeh, P, M., et al. [260]	2010	Race car design	Particle swarm multi-objective collaborative MDO, a fuzzy decision maker is used to select best design along Pareto front

One important part of the MDO process is in how the designers go about decomposing the original problem into a set of sub-problems. Decomposition can be seen as identifying weak links between elements that are coupled, and therefore allowing the elements to represent individual though coupled optimisation problems [213]. In general decomposition methodologies can be done in several ways such as object, aspect, sequential and model-based [218]. Model decomposition is a partitioning method based upon functional dependencies between design variables and functions included in the problem [249]. The main approaches to decomposition are the aspect-based and object-based methods, these form the next section of literature and include examples of MDO application to design optimisation based upon these decompositions.

3.2.2 Aspect-based decomposition

Aspect-based decomposition focuses on breaking up the particular problem based upon the actual discipline analysis associated with it as shown in figure 3.7. This can be aerodynamics, structural, thermal in the case of aircraft design, or electrical, mechanical, fluidic and structural in the case of a MEMS device.

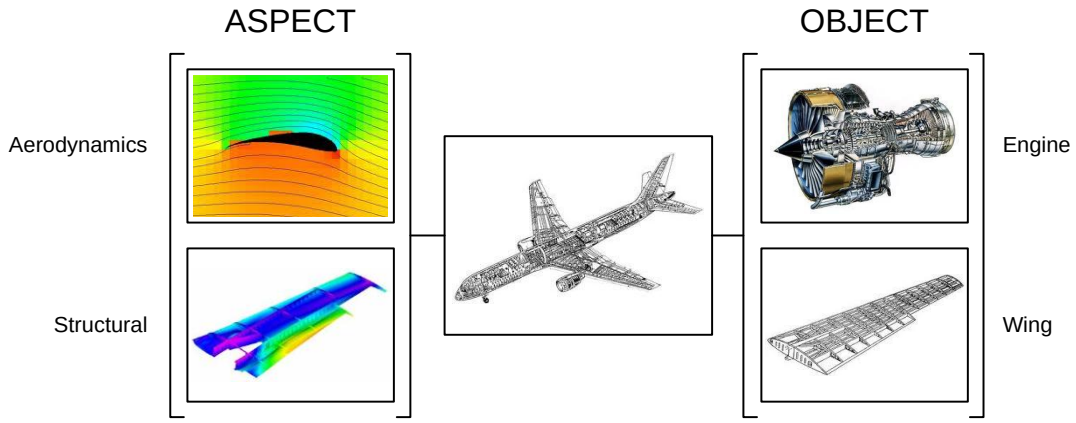


Figure 3.7 Decomposition of an aircraft based upon (a) aspect or (b) object methodologies

An example can be seen in [215] with the collaborative MDO of a supersonic aircraft design. Here the goal was to design a supersonic aircraft using three separate disciplines, aerodynamics, structures and mission analysis. The objectives were to minimize the take-off weight subject to typical mission performance and aerodynamic / structural constraints. Further examples of aspect decomposition within the literature are listed in table 3.7.

Table 3.7 Chronological review of developments in aspect-based decomposition

Contributors	Date	Applications	Description
Kroo, I., and Manning, V. [215]	2000	Supersonic aircraft design	Decomposition of supersonic aircraft into three major disciplines (aerodynamics, structures and mission analysis)
Giassi, A., et al. [261]	2004	Roll stabilizer fin	Sequential optimisation with hydrodynamic optimisation solutions fed into structural subsystem optimizer compared against MORDACE which provided superior performance
Tribes, C. [212]	2005	Structural wing	Decomposition into a system level performance objective and subsystem aerodynamics / structural disciplines
McAllister, C. D., et al. [253]	2005	Race car design	Consisted of two system level objectives, minimize lap time and maximize normalized weight, with subsystem decomposition into aerodynamic and force disciplines
Huang, H-Y., and Wang, D-Y. [259]	2009	Container ship	Decomposition into static, mode and dynamic disciplinary analysis
Zadeh, P, M., et al. [260]	2010	Race car design	Similar decomposition to [253] with aerodynamic and force disciplinary analysis

3.2.3 Object-based decomposition

In large-scale design environments the system as a whole can be structured according to the individual components of the system such as turbine engines or wing structures. These often correspond to engineering departments within a company and an object-based decomposition approach mimics this. Decomposing the problem into individual components brings with it a natural mirror to real-world design optimisation along with a simplification and grouping of design variables associated with these components.

An interesting example is found in [211] for the conceptual design of a bridge structure. The use of a collaborative MDO algorithm in which a bridge design is decomposed into two groups of optimisation, a superstructure subsystem and a deck subsystem is undertaken. Each subsystem has control over their own search and can choose from a number of different design concepts and variable parameters. These can include various superstructures (fan, harp or semi-harp suspension) or deck designs (steel plate girder, reinforced concrete box girder). Overall it was the first demonstration of conceptual design autonomy in a MDO problem, and the approach was successful in converging to an optimal solution. Further examples of object decomposition within the literature are listed in table 3.8.

Table 3.8 Chronological review of developments in object-based decomposition

Contributors	Date	Applications	Description
Balling, R., and Rawlings, M. W. [211]	2000	Structural bridge	Decomposition of the main components of bridge structure, the superstructure and deck in a conceptual MDO approach
Kurapati, A., and Azarm, S. [254]	2000	Speed reducer	The design problem objectives and variables are decomposed up into separate subsystems and solved independently before recombining
Gunawan, S., et al. [255]	2003	Speed reducer, UAV payload	Payload design with the goal to maximize probability of success, UAV design variables decomposed between subsystem levels
Aute, V., and Azarm, S. [257]	2006	Speed reducer, numerical test problem	Decomposition of design problem objectives and variables, similar to [255]
Rabeau, S., et al. [258]	2007	Speed reducer, dock design problem	Decomposition of dock structure into separate subsystems containing individual cantilevered beams attached to vertical wall

3.3 Summary

A literature review of the field of multi-level design optimisation was undertaken with a focus upon multi-level evaluation, parameterization and search strategies. Covering the major developments associated with each strategy and demonstrated through real-world examples, their application to MEMS design optimisation is a promising direction for investigation. Additionally a review of the literature was also undertaken on a similar area of research, multidisciplinary design optimisation. MDO is a class of coordination algorithms that look to decompose complex design problems into a number of subsystems to improve the design process. Such a strategy is also a suitable approach to MEMS design optimisation and warrants further investigation.

4

Design Optimisation Strategies and Framework

This chapter provides a brief overview of how the principles of multi-level and multidisciplinary design optimisation can be applied to MEMS design, along with a brief outline of the forthcoming multi-level and multidisciplinary experimentation undertaken in this thesis. The chapter then introduces the design optimisation framework environment used throughout the remainder of this thesis, firstly with a brief overview of the structure and control elements and then focusing towards how data is stored and interacted. The remainder of the chapter focuses upon the validation of the multi-objective evolutionary algorithms used throughout the thesis through two multi-objective and multidisciplinary design problems. The MDO proposed is also constructed within the computational framework and validated against the same set of design problems.

4.1 MEMS multi-level optimisation strategies

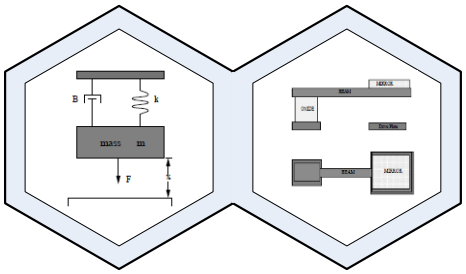
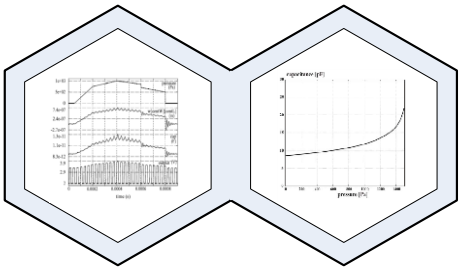
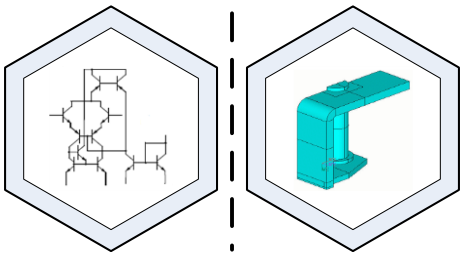
The more traditional stochastic methods such as simulated annealing and the more popular genetic algorithms have shown themselves to be successful and in some instances superior to more local based automated optimisation routines such as gradient search in MEMS design.

However given the complexity of MEMS design synthesis and optimisation any improvement that can be incorporated into the standard evolutionary algorithm routine to further aid design synthesis and optimisation is an important contribution. Examples of such include the integration of user interaction through interactive evolutionary computation methods [43], improved fidelity at the final stages of design through hybrid heuristics which couple both global search algorithms and local gradient based methods [124], and methods to facilitate more robust design procedures so as to insure final designs are as robust to fabrication errors as can be possible [123].

Another area of research is that of multi-level design and optimisation, a field of research which has shown great success in its application to macro scale engineering design problems over a variety of disciplines. This is often a result of the ability to exploit the large number of modelling and analyses tools available to engineers and develop strategies which reduce the complexity and overall computational cost of design optimisation. Also given the complex nature of most engineering design problems with constraints and large numbers of components and design

variables it can quickly become intractable to undertake such design optimisation using standard all-in-one methodologies. Multi-level strategies also provide a means to reduce the complexity of design problems in manageable hierarchical levels of design optimisation, and the ability to integrate both global and local search algorithms into a unified process is particularly useful for MEMS design given the scales involved and the effect small variations in design parameters can have on the often non-linear behavioural responses of a device.

Table 4.1 Multi-level MEMS design examples

Contributor	Approach	Comments
	All In One [Component + Component] / [Analysis + Analysis]	Behavioral model for a MEMS pressure sensor which automatically produces process level layout specifications depending on user supplied behavioral targets [109]. [106] Coupled system level circuit analysis and physical level FEM / flow analysis.
	Separate analysis process [Analysis] + [Analysis]	Black box modelling used to create artificial neural network of pressure capacitor than linked with circuit model for fault tolerance design [203].
	Separate design process [Component] + [Component]	Electrical equivalent model for the magnetic subsystem of a microrelay is optimized and used a base for further FEA verification after optimization is complete [204].

There are examples of the use of multiple levels of design and analysis, in particular through the use of different levels of modelling abstraction, within the literature of MEMS design optimisation. Given the nature of MEMS modelling and analysis, these examples often fall upon the use of multiple modelling abstractions or components [50][53][203][204]. One of the most common is the use of modelling and analysis from one or more levels of the MEMS design hierarchy. Table 4.1 highlights how a device level behavioural model is used to design the process level layout of a pressure sensor [109], along with a coupled system and physical level iterative design optimisation process for a flow sensor [106]. Other examples build simplified modelling and analysis representations from more complex and computationally expensive models in order to speed up the design process [203]. It is also often that the design process is broken up into two separate design synthesis and optimisation cycles which contain different

components, models or analysis tools. In [53] a methodology for design of a bandpass filter crossed two different levels of modelling and analysis, in this instance a system level design through the use of bond graphs and device level design through the use of a parameterised analytical model. In this instance each modelling level occurred in separate design processes, with the final optimal design of one level providing objective input for the other. In [204] rather than use expensive physical level FEA during the design optimisation of component for a microrelay, a system level electrical equivalent circuit model representation was used instead. Once the device was optimised the geometrical values for the device were extracted from the electrical equivalent model and used to build and verify the final FEA model.

Though multiple components or modelling abstractions are utilised in examples throughout the literature, they are often *ad hoc* and utilise the various models and analyses in an all-in-one representation or in separate optimisation cycles. The application of tailored multi-level design synthesis and optimisation strategies akin to what has been described in earlier chapters is not so withstanding, which is surprising given the possible benefit a number of strategies could play in the overall process of MEMS design.

MEMS design synthesis and optimisation encompasses a number of possible multi-level strategies that could be applicable and beneficial to the improvement of the design process. The following sections outline the multi-level design synthesis and optimisation strategies that are to be employed within this thesis in more detail. Already discussed three multi-level strategies stand out amongst all others, multi-level evaluation, multi-level parameterization and multi-level search.

The availability of various modelling and simulation abstractions and the often prohibitive nature of high cost CAD/CAE in the final stages of automated design make the attraction of multi-level evaluation strategies to the application of MEMS design high. Any opportunity to increase the number of functional evaluations available to a designer when undertaking automated design synthesis and optimisation of MEMS is beneficial and therefore the first section focuses upon how this can be achieved using a multi-level evaluation strategy for MEMS design.

The increasing complexity of MEMS design synthesis and optimisation, in particular the relationship to increasing components and their related design variables of larger devices means that even the more powerful stochastic automated optimization algorithms will only begin to struggle as they get larger and more complex. Multi-level parameterization strategies look to reduce the complexity of a problem by simplifying the search space or opening it up to larger search by reducing the number of design variables present or essentially simplifying the choice an optimisation algorithm has to make when looking to optimise a device. Following on from multi-level evaluation strategies a look into how multi-level parameterization strategies can be applied to the MEMS design process is outlined.

The final multi-level strategy discussed, multi-level search, is not undertaken within this thesis toward MEMS design synthesis and optimisation for a number of reasons. Firstly the focus of optimization is upon the field of evolutionary computation and in particular the use of multi-objective genetic algorithms and therefore the use of additional local gradient based algorithms are not explored. Within the literature there has also been extensive investigation into a similar

area of research, hybrid optimisation algorithms for MEMS design synthesis and optimisation [42][124], and with the need to maintain some level of brevity within this thesis only two of the three multi-level strategies are experimentally explored.

4.1.1 Multi-level evaluation for MEMS design optimisation

Exploration and exploitation are common features of evolutionary algorithms where the fluctuating dynamic between trying to explore the search space during the design process to find suitable high fitness regions and then exploiting designs within this region in order to more rigorously search local fitness landscapes for an optimal solution. Such exploration however often involves a large number of functional evaluation calls and therefore it is beneficial to a designer to be able to increase this number as much as possible within the available time frame.

Interpreting the scope of multi-level evaluation strategies for MEMS design optimisation often falls into the means at which a solution is evaluated and assigned a value or fitness indicator. The main attribute is the balance between evaluation accuracy and cost between levels and how the trade off from a low accuracy but lower cost evaluation software and a more costly and accurate one. Cost is often associated with wall clock time, or simply the amount of time required carrying out an analysis / simulation of the chosen component. However the cost can also be evaluated in terms of resources such as memory, where often expensive CAD/CAE which utilize finite element analysis carry a much larger memory footprint then other simpler analytical methods.

MEMS modelling and analysis is split between numbers of hierarchical levels each containing tools which exhibit a range of computational cost and analysis accuracy. Often the most accurate modelling and analysis tools involve computationally expensive CAD/CAE methods which make it time-consuming and in some instances impossible to use in an automated iterative optimisation routine such as those found in evolutionary computation. This is problematic as designers often use such tools for their increased accuracy and range of disciplinary analysis.

One particular class of evolutionary algorithm tailored towards assisting computationally expensive design optimisation such as that found in engineering industries which use CAD/CAE software is the so-called metmodel-assisted evolutionary algorithms (MAEAs)[205][206][46]. The ability to screen out non promising solutions before committing to more expensive exact evaluation or increase the number of functional evaluations by utilising lower cost metamodels is of great benefit in design fields of limited resources or time, or where its reduction is economically advantageous.

The most common surrogate modelling techniques are multilayer perceptrons [207], radial basis function (RBF) networks [207][208], response surface methods [209] and the Kriging techniques [210]. MEMS modelling and analysis however contains a number of abstract levels in which to extract suitable models for simulation when available without the direct need to use such surrogate approaches. Therefore is it possible to perform suitable multi-level evaluation strategies to increase the number of functional evaluations during the automated design process using the modelling and analysis tools available without the need to build surrogate models.

MEMS are easily suited towards multi-level evaluation strategies given its hierarchical design process and the need to improve automated design optimisation wall clock time. Abstract and reduced order modelling approaches are becoming more common within MEMS design and there use allows for faster simulation and analysis though at a reduced level of accuracy. Linking the two alternative methods of analysis allows for increased exploration of the search space at a similar or reduced cost in comparison with simply performing design optimisation using a single high cost method.

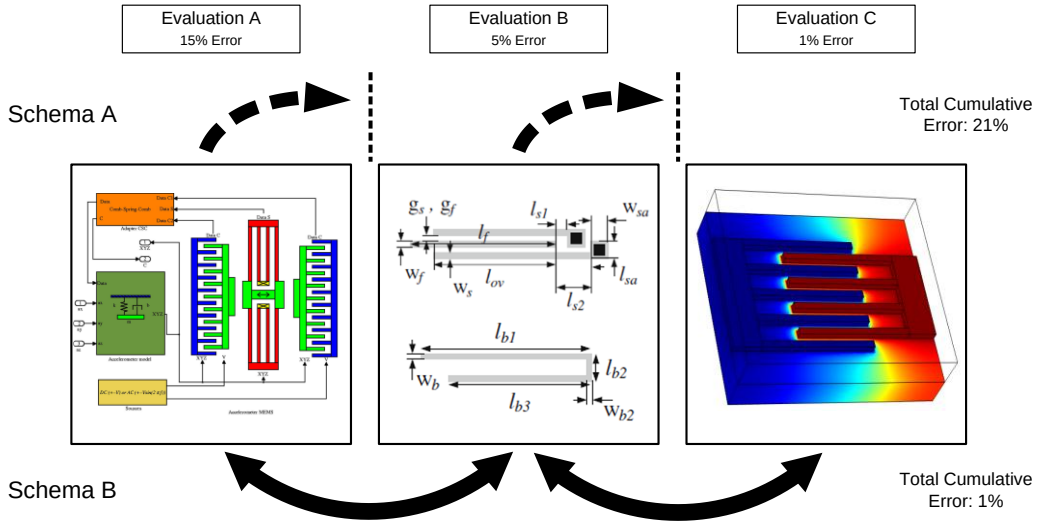


Figure 4.1 MEMS inter-level evaluation

The hierarchical design process has been broken down into a number of levels at which a designer can model and analyse a device. The system, device and physical levels outlined by Senturia [16] cover the majority of modelling representations used within automated design synthesis and optimisation of MEMS. There are two possible directions in which a designer can look to employ a multi-level evaluation strategy within the framework of these three levels. The first is outlined in figure 4.1 and involves two or more levels of modelling and analysis, in this instance system, device and physical level models for a MEMS comb transducer. The multi-level evaluation structure revolves around each of these separate levels with each one associated within a separate deme or population set, or called sequentially within a single population. This approach can be seen as an 'inter-level' structure within the MEMS hierarchy of modelling and analysis tools as it involves abstractions from more than one level. Careful consideration should also be given in how each level is linked within the design optimisation procedure. In figure 4.1 schema 'A' undertakes what is a de-coupled optimisation procedure similar to [50][53] where one level of modelling and analysis is used during optimisation to build further design targets for other levels of modelling and analysis. It is possible for error to carry from one level to the next and accumulate within the final verification at the more expensive physical level. Schema 'B' in figure 4.1 involves integrating all modelling and analysis levels within a single design optimisation routine. Here no design targets are transferred with only the effects of error from each level remaining local and final design need only be taken from the most accurate level of modelling and analysis.

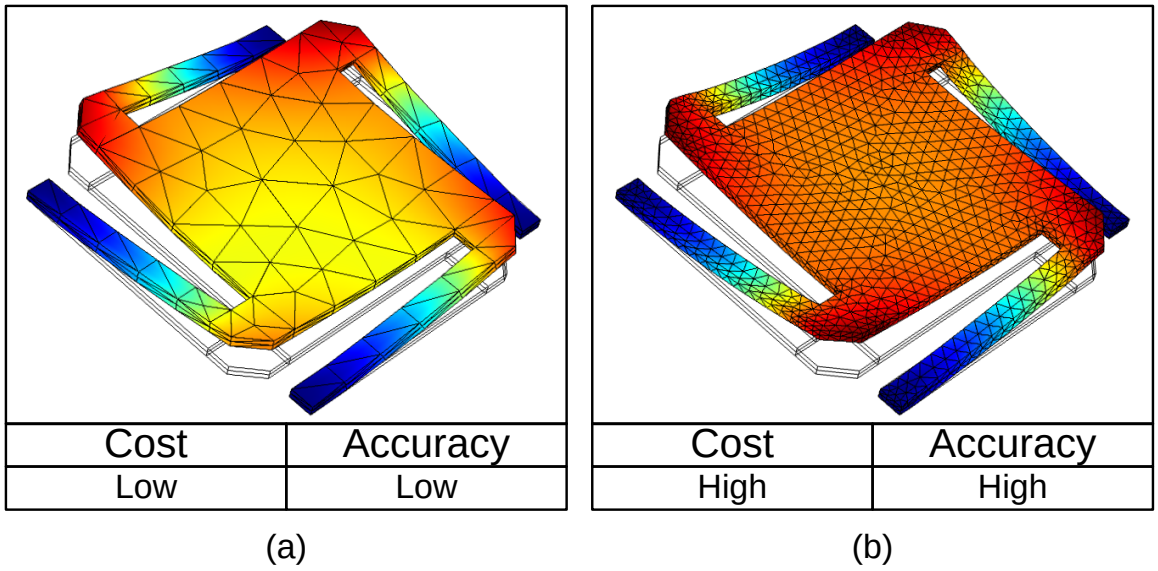


Figure 4.2 MEMS intra-level evaluation with (a) course mesh FEA model and (b) fine mesh FEA model

The second possible direction for multi-level evaluation strategies for MEMS design involves applying them within a single or 'intra-level' only as shown in figure 4.2. In this example a MEMS micro-actuator is modelled at the physical level using FEA in both a low cost and low accuracy model (a) and a high cost, high accuracy model (b). This is can be achieved by altering the meshing of the device model during its construction to give both course or finely meshed models. Other examples may include coupling analytical modelling [147], itself relatively fast and accurate if not restrictive, with another device level tool of NODAL modelling and analysis [66], itself slower but containing more fidelity in design.

The drawback of the application of a multi-level evaluation strategy is that abstract models need to exist in order for such a method to be available and useful and by reducing accuracy there is a chance that search can become ambiguous and transferring between levels can escalate this. Therefore it is essential that any case study investigating the role multi-level evaluation can play in MEMS design synthesis and optimisation cover all three levels of MEMS design and analysis.

Careful consideration must also be given into assessing the cost of the models and analysis used during the design process for each level. In a standard single level optimisation approach the total cost is applicable to the number of evaluation calls put forward to the analysis software, often of the highest fidelity, for example CAD/CAE in engineering. The average computational cost of analysis in time can therefore be used as a basis for determining how much a single level functional call costs relative to less accurate but cheaper models [182][188]. In a multi-level evaluation approach this cost can be apportioned among the levels in such a way that comparatively both remain equal, and as such an accurate determination of cost and its ratio between low and hi fidelity analysis calls needs to be determined. The term 'cost' is also subjective and often depends on the largest limiting factor when it comes to the design optimisation process. The cost can be equated to analysis call time, computational resources such as memory or access to grid computing.

4.1.2 Multi-level parameterization for MEMS design optimisation

Complexity is a topic of interest within the field of computer science, as technology grows, as it is incorporated more into the lives of everyone, as tasks and functions are undertaken by ever more complicated devices, overcoming complexity and its problems becomes more paramount.

In the area of design optimisation, particularly within engineering, complexity can stifle or be pathological with regards to automated methods designed to aid the optimisation process [182][195]. The reduction of complexity so as to give rise to a positive effect on design optimisation is important if industries such as MEMS design are to be opened up to a wider audience such as application designers [21].

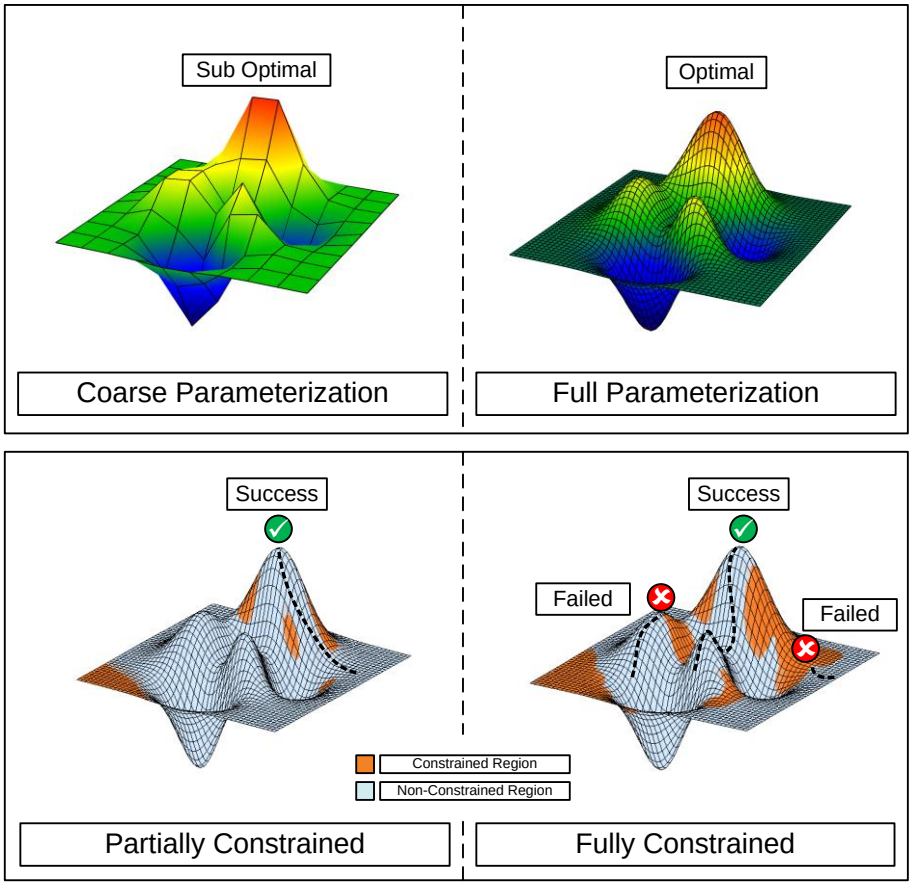


Figure 4.3 Multi-level parameterization of design variables and constraints

The overall main theme of multi-level parameterization is the reduction of complexity through the use of varying levels of model or problem granularity as shown in figure 4.3. Here coarse model representations are used to reduce the possible 'load' or dimensionality on any search algorithms employed in the design process, while the higher levels often employ a complete representation which opens up the search space comparatively for a larger range of possible search directions. In this top example the course parameterization and the full parameterization give rise to two different dimensional search spaces for the chosen algorithm to work on. It is also possible to include varying levels of active modelling constraints for design problems with restrictive constraints that make it hard for an optimizer to evolve solutions that remain feasible and unconstrained.

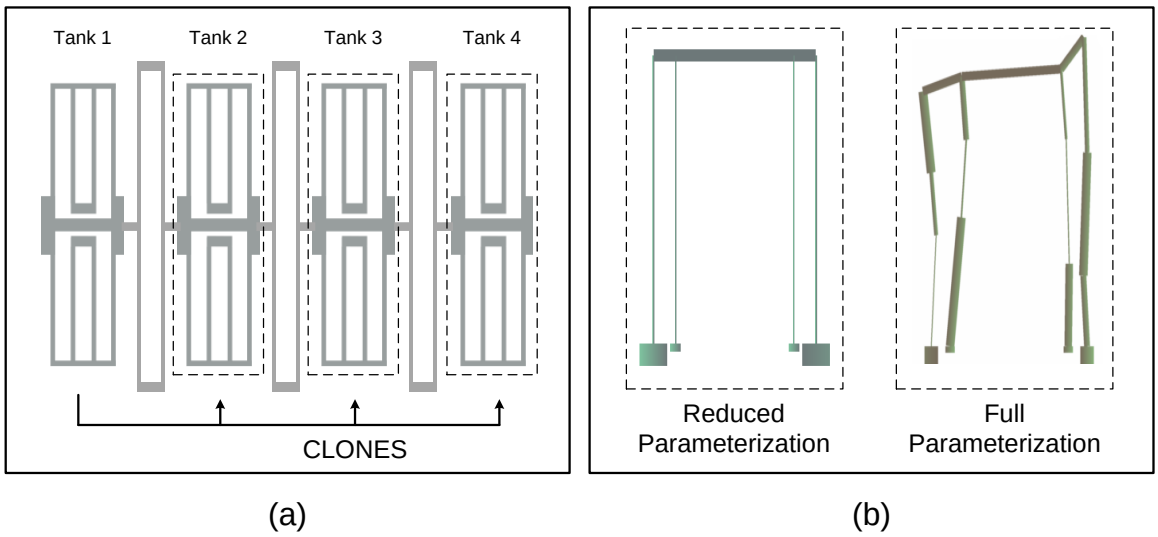


Figure 4.4 MEMS multi-level parameterization through (a) cloning of component design variables or (b) mixed levels of parameterization

Microelectromechanical systems or devices can consist of multiple components that contain a large number of design variables. An example of a complex system of components working in unison to perform some level of cooperative function would be a multi-stage bandpass filter [50]. Figure 4.4a shows a simple outline of such a bandpass filter built from a series of folded flexure resonator and coupling spring components. The application of a multi-level parameterization strategy may therefore look to reduce the complexity of designing such a system by reducing the number of design variables open to the optimizer. At the lowest level only one of the folded flexure resonator 'tanks' need be parameterised while all remaining tanks can simply be clones of the first. The highest level can than consist of the full set of design variables for each folded flexure resonator model within the bandpass filter allowing the optimizer to explore the search space further. Another example of the application of multi-level parameterization is shown in figure 4.4b with the modelling of the folded flexure spring component of a single folded flexure resonator. It is possible to parameterise and constrain such a component so it consists of a very simple and rigid design or open it up into something more complex.

4.2 MEMS multidisciplinary optimisation strategies

Deconstructing or decomposing complex engineering problems as often found in modern real world industries into simpler sub problems spread throughout a hierarchy of levels can prove to be computationally advantageous [152]. In large design problems the number of design variables and constraints can become prohibitive with regards to direct optimisation and as a result give rise to intractable problems. By reducing the whole problem into more manageable tasks which can be solved independently such a design problem can be approached more easily than would otherwise be possible and opens them up to parallel or multiple computer processing, thereby shortening the design cycle time and cost [152][170]. Expanding the current state of the art algorithms in automated MEMS design, the standard multi-objective evolutionary algorithm, to incorporate such decomposition and coordination techniques could therefore be very advantageous.

Early work on decomposition of multi-objective problems can be found in [262] where the coordination of a global objective often the optimisation of a large scale system is structured into a two tiered system and subsystem level. Here a multi-level approach controls the overall global and subsystem optimization, with each subsystem able to control a single objective of a multi-objective problem while the system level coordinates everything towards a global set of solutions. A number of decomposition based methods have also been developed within the field of evolutionary computation [263][264] and have shown an effective reduction in search cost and improved design performance as a result.

The bulk however of multi-level approaches which decompose the primary problem statement into a system level design problem and a set of uncoupled or coupled component level problems are primarily found in the field of coordination algorithms coined multidisciplinary optimization (MDO) [222-226].

The field of MDO includes a number of multi-objective population based approaches, the most successful incorporating a bi-level structure similar to [262] which looks to coordinate a number of subsystems each with their own set of populations and optimizers. The current state of the art in multi-objective MDO which integrate multi-objective genetic algorithms into the architecture of the MDO routine is the found in the work performed by Azarm, S. *et al.* [254][255][257].

The next section looks at how this particular approach to multi-objective population based MDO can be evolved to work within the field of MEMS design synthesis and optimisation through the creation of a novel non-hierarchic MDO algorithm.

4.2.1 Architecture of multidisciplinary optimisation for MEMS design optimisation

Microelectromechanical systems often contain a large number of coupled devices or components that provide some form of desired behaviour or function through their collective actions. The system as a whole or the individual components that make it up also often covers a number of disciplinary domains, be they mechanical [3], electrical [50], or more recently fluidic [5] and biological [7]. The increased complexities from designing such multidisciplinary systems can make it harder for designers to build such devices as they often require explicit knowledge in more than one discipline. The application of automated design synthesis and optimisation techniques towards multidisciplinary design problems such as those found in MEMS could greatly speed up the design process and ease the burden of design placed upon the designer.

The relationships between the disciplines or components within a design problem often form the basis for the structure the multidisciplinary optimisation routine will take when looking to apply a MDO algorithm. The current state of the art in multi-objective population based MDO employs a multi-level hierarchical structure with an upper and lower level relationship that can be structured to contain the decomposed design problem into a set of discipline or component subsystems as shown in figure 4.5.

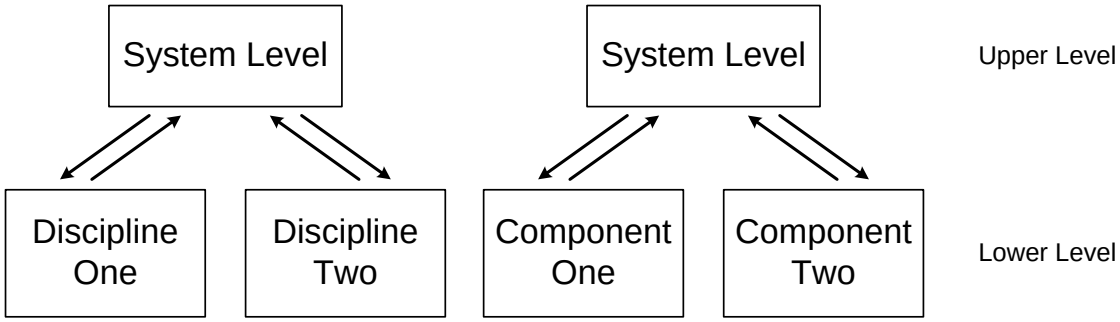


Figure 4.5 Multi-level hierarchical structure of decomposed MDO design problem

These two approaches mimic the aspect and object decomposition methodologies described previously and both can be equally applied to the MDO of MEMS. Figure 4.6 provides an example of how a real world MEMS device, the ADXL150 accelerometer, can be broken up using an aspect based (a) and an object based (b) methodology. Here the aspect based decomposition contains lower level subsystems which undertake specific disciplinary analysis required for design optimisation with design variable, objective and constraints often linked to the individual discipline. The object based decomposition concerns its self with the major constituents of the device or system, with design variables heavily linked to these constituent parts and objectives and constraints often tailored so as to optimise these individual components in such a way as to benefit the global design goals situated at the system level.

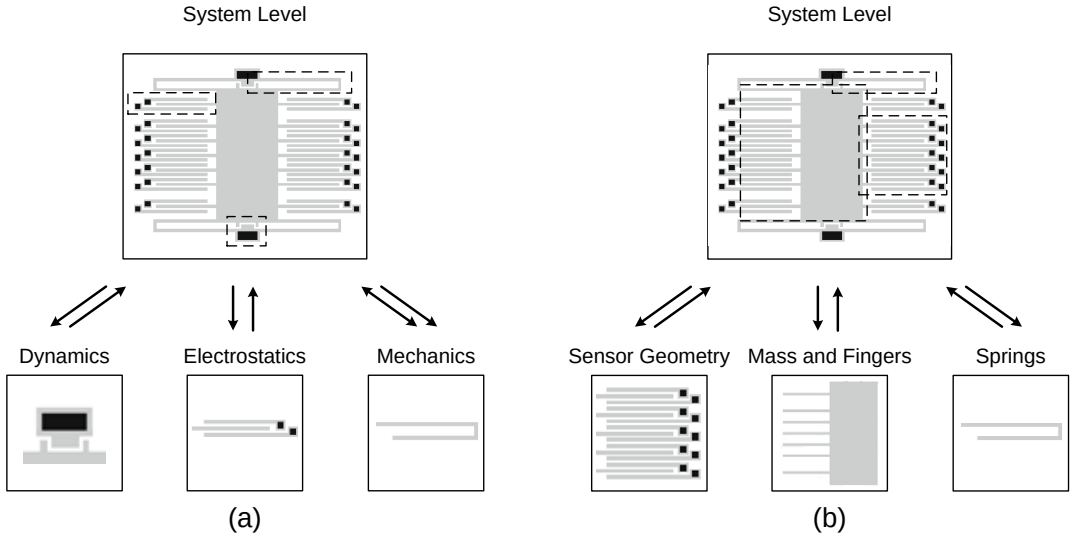


Figure 4.6 Decomposition of ADXL150 accelerometer for MDO using an aspect based (a) or object based (b) methodology.

The integrated and coupled nature of MEMS and the devices and components within them can mean that it is not always possible to fully decompose a design problem and that there still requires some level of communication between each of the lower level subsystems. In the ADXL150 accelerometer example outline above, it is conceivable that analysis and design variable information altered within one subsystem is needed by another. The calculation of the

electrostatic force is required for the calculation of the mechanics of the device in particular the displacement and stiffness of the suspended springs [40].

The current state of the art in multi-objective population based MDO employs a hierarchical structure as shown in figure 4.7a with each individual lower level subsystem isolated from all others in a fully decomposed design problem. Such hierarchical structures often require the design problem itself to be hierarchically decomposable with its objectives separable or additively separable which may not always be possible [255]. A non-hierarchical structure as shown in figure 4.7b allows communication between the individual subsystems therefore allowing solutions within each subsystem to be provided with the correct disciplinary analyses or subsystem design variables.

A number of ways have been presented on how to transfer coupled variables in order to reconcile each of the subsystems into the formation of a complete solution. The cooperative co-evolutionary algorithm set out in [167] looks to choose the current best solution from each sub species and recombine them with the chosen solution in the current subsystem to be evaluated. In [258] a different approach looks to pass approximations of coupled variables from the system level to each subsystem. The difference between the real, but inaccessible, value and the approximate values decreases during the optimisation process. Updated coupling values from each subsystem are sent at every system level invocation and then passed on to all other subsystem levels later on, however they soon become approximations again as each subsystems optimisation routine evolves.

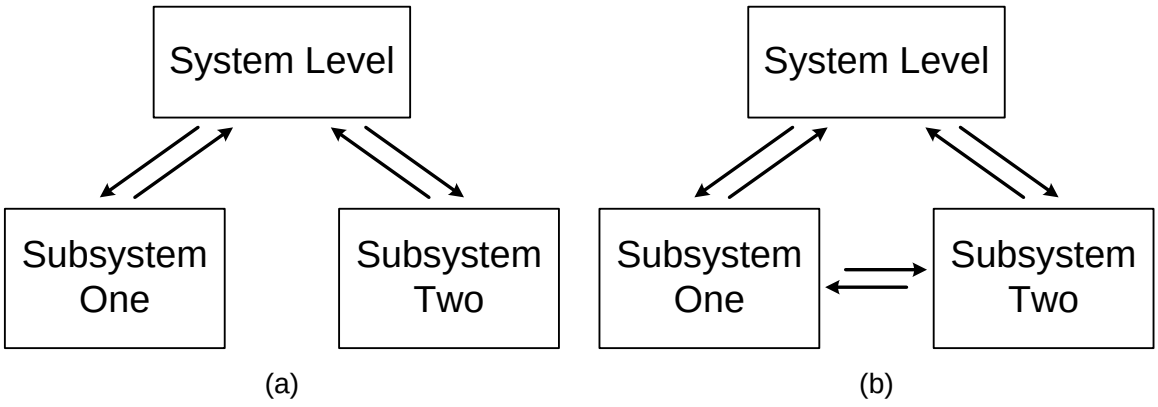


Figure 4.7 Hierarchical (a) and non-hierarchical (b) multi-level MDO structural relationships

4.2.2 Multi-level and multidisciplinary experimentation formulation

Outlined previously was an overview of the fields of multi-level and multidisciplinary optimisation with a particular focus upon population-based optimisation algorithms. The principles of both approaches were also discussed with regards to their application to MEMS design synthesis and optimisation. Below is a presentation of how each approach, both multi-level and multidisciplinary optimisation, will be applied experimentally throughout the remainder of this thesis.

MEMS design synthesis is well suited to hierarchical design due to the many levels of modelling and analysis available to the designer, the increasing complexity of new devices and their design search spaces and the multi-disciplinary nature of MEMS themselves.

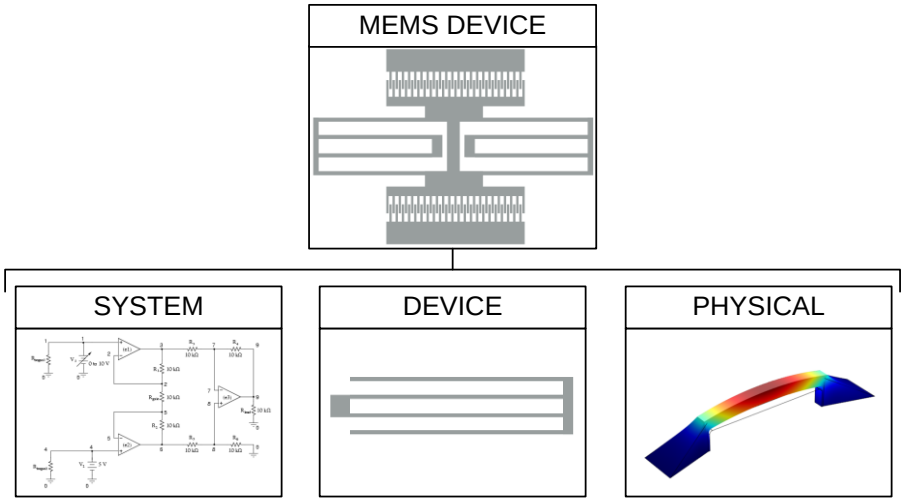


Figure 4.8 Hierarchical partition of a typical MEMS device: consisting of elements from all three abstract levels.

Outlined previously the hierarchical nature of MEMS can be partitioned into four separate levels, system, device, physical and process, each encompassing their own set of modelling and analysis tools. In a 'Top-Down' approach, MEMS design synthesis and manufacture can be divided into two main areas of broad work, the first focuses on MEMS design and function, where the designer or design team take the customer requirements set out and then look to create a device to meet these targets, be it by a hand driven, 'build and break' design process, or a more automated soft computing one. Once the design itself has been created the second task is fabrication of the device itself and outlining the process steps required to build it to the correct specifications. Naturally there is no clear divide between the two sections of work, and there is cross talk between them, decisions made at a design level have to reflect realities at the time of fabrication and processing. However the main focus of this thesis revolves around the design stage and therefore excludes the process level, leaving the three other levels of interest in MEMS design, the system, device and physical levels.

Customer requirements be they through the function of the device or some form of constraint on the system can also dictate how a component is modelled and analysed, or it may be simply advantageous to utilise certain modelling techniques as they open up design, for example shape optimisation of a FEA model at the physical level.

Therefore any case study that is to be used to investigate multi-level design optimisation of MEMS must also look to include some level of representation of the various modelling levels present in the hierarchical design process as shown in figure 4.8.

The outlined multi-level and multi-disciplinary optimisation methods can be applied in two distinct ways over the hierarchical MEMS design process. The application at a single level of the design process, coined 'intra-level' optimisation, involves the application of the proposed methods at an individual and isolated level, and therefore the various level abstractions or

partitions must occur within a single level only, be it system or device. The second application involves the application of each method over more than one level in either a bi or tri level arrangement, utilising two or more of the various levels, for example a system-device coupling, or a complete system-device-physical coupling design problem.

Therefore experimentation considers a look at all three levels of modelling and analysis, the first focusing on each level individually as shown in figure 4.9 with multidisciplinary optimisation and multi-level characteristics built from within each level only in what is coined 'uni-level' design optimisation. The next step upgrades experimentation to consider a MEMS design problem which couples more than one level of modelling analysis and design. Shown in figure 4.10 a set of 'bi-level' MEMS design problems are constructed with a coupled system and device problem and a separate physical and device level design problem. Once again the characteristics of both the multidisciplinary optimisation and multi-level algorithms are built from these two sets of coupled modelling and analysis levels. The final MEMS design problem shown in figure 4.11 encompasses all three levels of modelling and analysis in a complex 'tri-level' MEMS design problem looking to test both the current state of the art and proposed multi-level and multidisciplinary algorithms to their fullest.

Before experimentation can begin however two objectives still remain, the construction of a computational framework to hold the multi-level and multidisciplinary optimisation algorithms which are to be applied to MEMS design, and finally a MEMS multi-level case study suitable enough to evaluate both approaches. The next section looks to develop and outline the computational framework used throughout the remainder of this thesis, and follows this with a construction and validation of the multi-objective and multidisciplinary optimisation algorithms used against current state of the art examples.

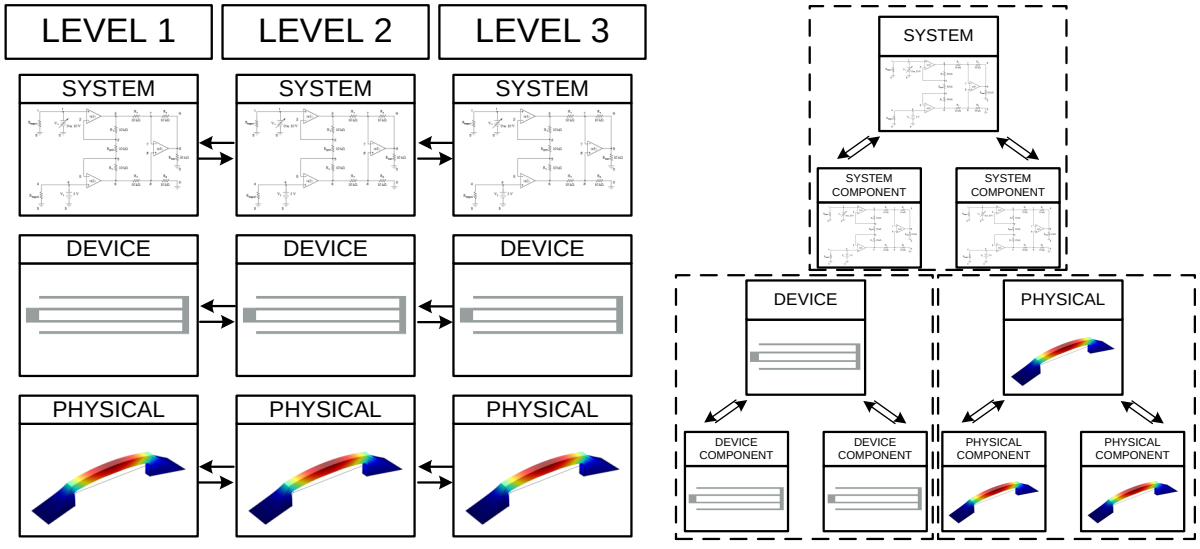


Figure 4.9 Uni-Level design optimisation case study breakdown

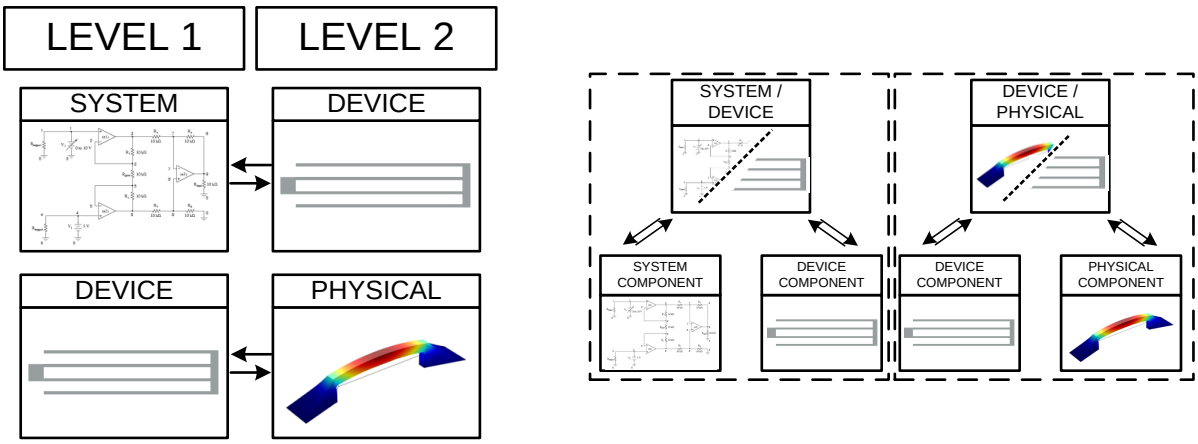


Figure 4.10 Bi-Level design optimisation case study breakdown

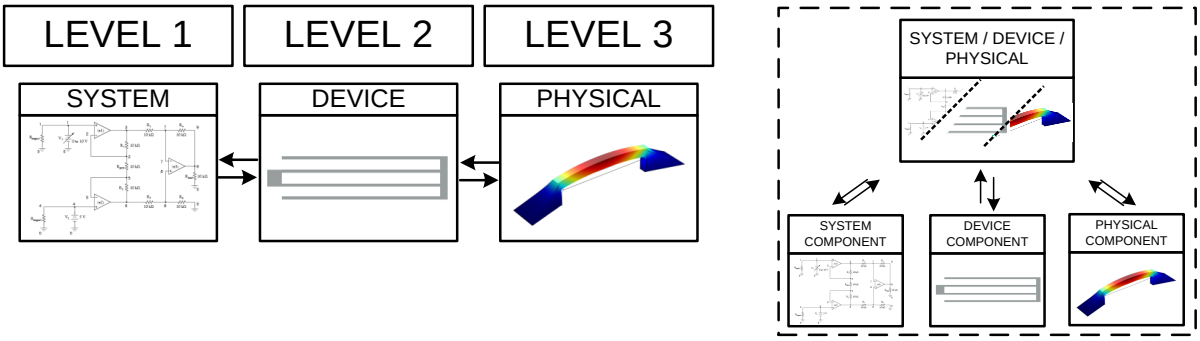


Figure 4.11 Tri-Level design optimisation case study breakdown

4.3 Modular framework for multi-level and multidisciplinary optimisation

The evolution of automated design optimisation of MEMS has brought with it a number of improvement over the course of a decade in particular with regards to the platforms and frameworks used to synthesize and optimise such devices. A large bulk of this has been driven from the work undertaken by the Berkeley group [43][124][125][137] through successive doctoral and academic researchers in the field of MEMS design.

The earliest work by [137] updated the traditional automated method for MEMS design optimisation to the more powerful stochastic evolutionary algorithms in particular those that focus on multi-objective problems common in engineering and MEMS design. MEMS devices and components that often make them up can be represented as individual building blocks or structural segments which can be connected to form larger devices. The work by Zhou took the standard representation found within genetic algorithms, that of a string of decision variables, and updated it to an encoding more suitable to MEMS design with the ability to construct data structures which can describe any number of MEMS building block components. It is particularly useful for exploring conceptual design synthesis as it no longer requires the use of a parameterized template to evolve optimised designs. The encoding for the MEMS devices within

the framework was once again expanded to include the concept of shape grammars [266][267] by Kamalian where rules could be coupled with set structural elements to create a process for evolving new designs from simple templates by add and varying the components that make them up. Kamalian [43] then explored the role of human interactive evolutionary computation, incorporating into a framework the ability for MEMS designers and their expertise to direct and control the evolutionary algorithms tasked with designing and optimizing the MEMS devices. Over time more improvements have been made through the addition of hybrid algorithms which contain both global and local searching capabilities [124], case based reasoning [125] and a hierarchical component based MEMS representation [42] to name but a few.

Any framework that is to be designed should look to have as many of the previous improvements as possible or in the very least have the ability for their incorporation at a later time. In particular careful consideration must be given into the representation that is used to store data in this case the decision parameters associated with the MEMS device or components. The framework also needs to be able to undertake automated MEMS design optimisation using multi-objective genetic algorithms, able to handle the complex structural methodology of multi-level and multidisciplinary optimisation outlined in chapters 3 and 4 for any number of design problems. Also given the highly disciplinary nature of MEMS, in order to evaluate design optimisation at all levels of the MEMS design process open to a designer there is a need to be able to incorporate modelling and analysis tools for the system, device and physical levels within the hierarchical design process.

The previous design optimisation framework was built using a design optimisation platform called modeFrontier [145] and tasked with optimising a number of multi-objective MEMS design problems using state of the art evolutionary algorithms NSGAII and MOGAII. Unfortunately the platform is a commercial tool with little scope for expansion into the various fields outlined previously and more importantly while it contains a simple array representation it is unable to handle multi-level and multidisciplinary optimisation. The remaining sections of this chapter outline the main components of a multi-level and multidisciplinary optimisation computational framework that has been constructed to undertake the experimental analysis of the proposed multi-level and multidisciplinary strategies.

4.3.1 **Operation principle of framework software**

The modular framework is built using an Integrated Development Environment (IDE) and implemented in C# [268] programming language, an object-orientated language developed by Microsoft and part of its .NET initiative [269]. The framework is structured into three main sections:

1. Framework structure and control
2. Framework module components
3. Solution representation

The overall modular framework can be seen as a generic modular ‘black box’ system which has been used to build an automated design optimisation platform tailored to multi-level and multidisciplinary design. However it is open to any number of configurations, for example agent-

based or interactive design optimisation provided the suitable modules are built within the framework.

4.3.2 **Framework structure and control**

The modular framework is designed towards a hierarchical structure with a number of levels of command and control. This autonomy is both local, with control over levels below the hierarchy, and also dispersed with the ability for levels to communicate over the entire framework. The goal is to allow the construction of simple or complex heuristics in a structured and controlled environment with a specific tailoring towards automated design optimisation.

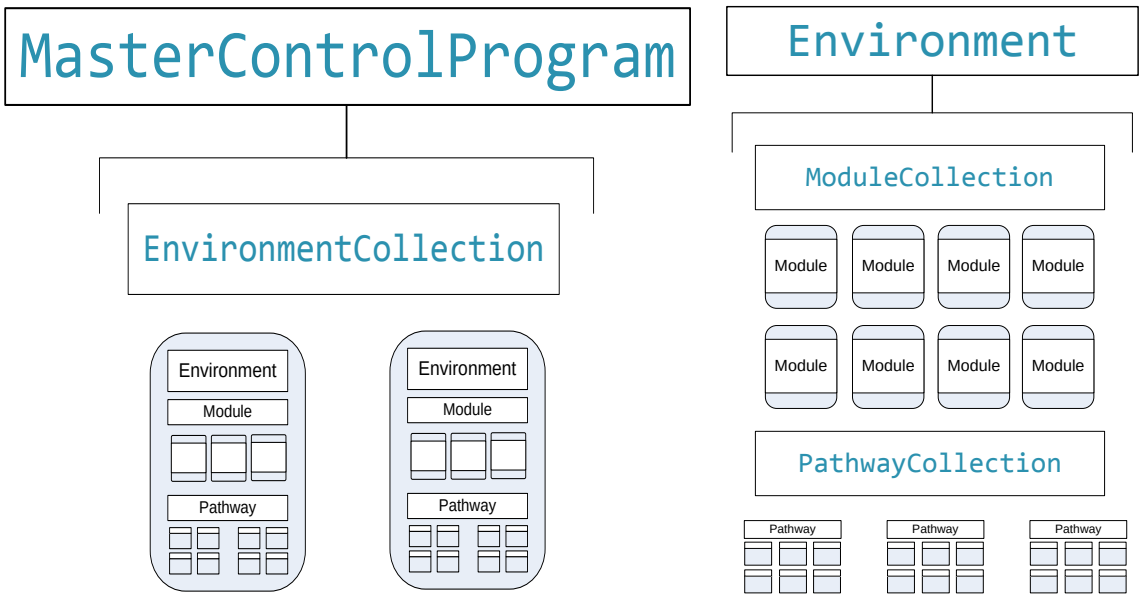


Figure 4.12 Master control program (MCP) object scope (left) and framework environment objects (right)

Figure 4.12 outlines the *master control program* (MCP) object which is the highest level of control for the framework and holds all environment objects created by the user. Communication between each *environment* is handled through the MCP, and the overall '*cpu cycle*' control is undertaken at this level. The main area of function processing is undertaken and overlooked by the *environment* object within the framework as seen in figure 4.12. The *environment* object contains two areas of interest, the *module collection* object and the *pathway collection* object. The *module collection* contains all modules for which this particular *environment* object has control and scope, while the *pathway collection* holds all *environment pathways* in which this *environment* object has control and scope over.

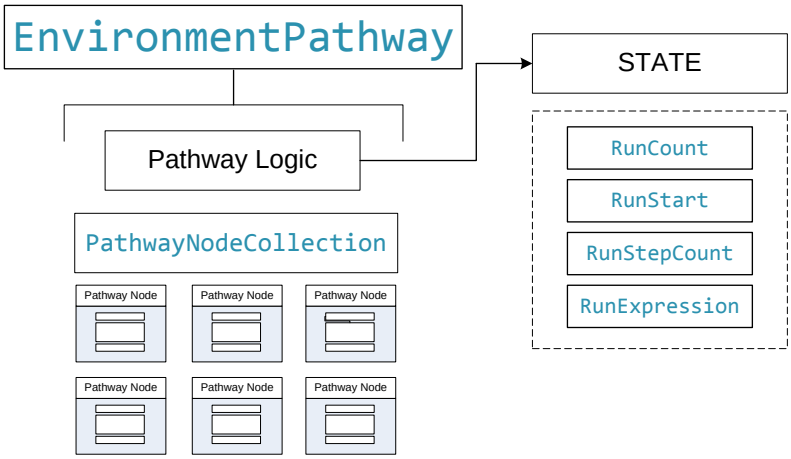


Figure 4.13 Environment pathway object

Within each *environment* there can contain a number of *module* objects for which an *environment pathway* can be constructed around as shown in figure 4.13. *Modules* within the framework act as the main centre of work or function, acting upon both inputs and outputs as simple black box object. The *environment pathway* acts as a linear nodal path of *modules* which allows a number of *modules* to act together to form a larger whole and or process. The coupling of each *module* is undertaken by a wrapper object called a *pathway node*, which is held internally in the pathway node collection object. The *environment pathway* also contains a *pathway logic* object, which acts as a higher control on the state of the *pathway* and allows for the *pathway* to be called or paused in numerous situations. The variable '`RunCount`' holds information on how many cycles or ticks a given *environment pathway* is called sequentially, the '`RunStart`' variable simply holds information on what cpu cycle or tick the *pathway* is started while the '`RunStepCount`' variable holds information on the number of cpu cycles or ticks that must be passed before updating the current state. There is also a higher level of control provided by the '`RunExpression`' variable which can be used to hold more complex logic with a true/false output to decide whether the *pathway* is invoked or not, this variable is null by default.

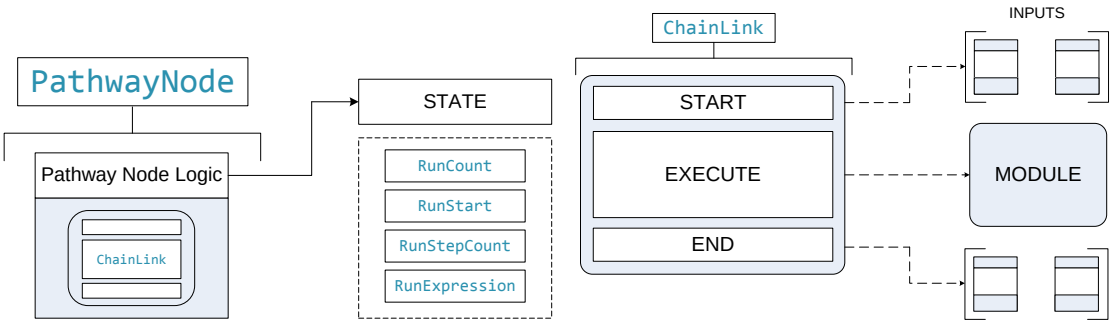


Figure 4.14 Pathway node object (left) and chain link object (right)

The *pathway node* object shown in figure 4.14 acts as a wrapper and holds a reference to the *module* object it contains. It communicates and is controlled by the *pathway* itself however it also contains its own *pathway node* logic. The *chain link* object in figure 4.14 acts as a second level wrapper for the *module*, holding 'input', 'output' and 'use' module data it has the task of

retrieving and sending all input and output data, along with the overall execution of control of the module to accomplish its task.

The framework is designed to be accessible without the need to access the code itself and is partitioned into three general areas of interaction as surmised in figure 4.15. At the highest level a user need simply to provide new configuration file information to solve a particular problem at hand, given the required *modules* are already implemented. The configuration file allows control of the *environments* that are to be built, the *modules* within them and the *environmental pathways* that can be constructed. Access to *module* parameters, *pathway* and *node logic* are also available along with numerous other features necessary to build a heuristic to solve a particular design problem. The second level focuses upon two areas, the creation of new *modules*, which are designed to perform some task dependent on inputs and outputs, or the creation of new *data* objects, the main unit of information, for example the creation of new chromosomal representations. At this level there is currently a need to provide some level of programming in order to alter or construct new *modules* or *data* objects however this is guided with templates and a very structured outline within each of these components. The final level is the most complex and overreaching and contains core framework objects that go beyond the scope of most users looking to utilize the framework.

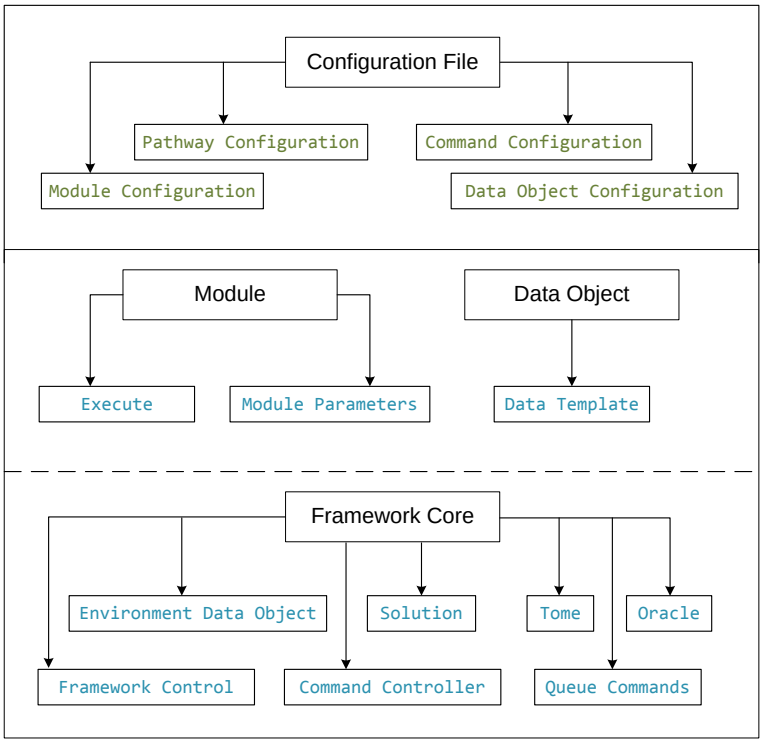


Figure 4.15 Hierarchical user control and input for modular framework

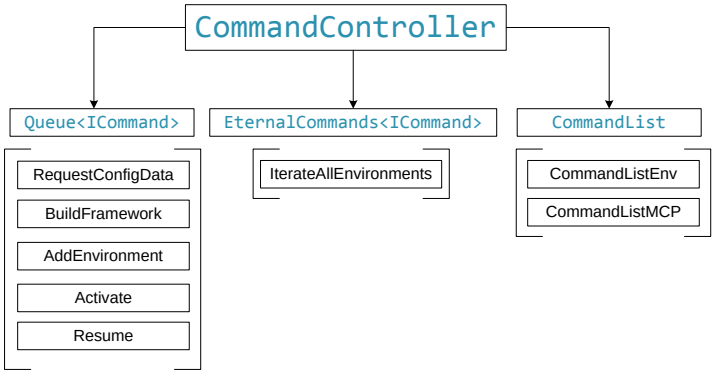


Figure 4.16 Command controller object

The framework contains a command and control strategy element with the ability for command packets or objects to be communicated between all areas of the framework, and much like a normal network placed within queues for invocation. This allows autonomous and dynamic control and function to occur throughout the framework without the need to program 'a priori' and therefore gives rise to the possibility of processes such as agent based control. Figure 4.16 shows some of the main commands and command lists within the framework, here queue commands are added and removed after they are called, while eternal commands remain within the queue indefinitely, here for example the '`iterateAllEnvironments`' command tells the framework to process each environment stored within the environment collection object.

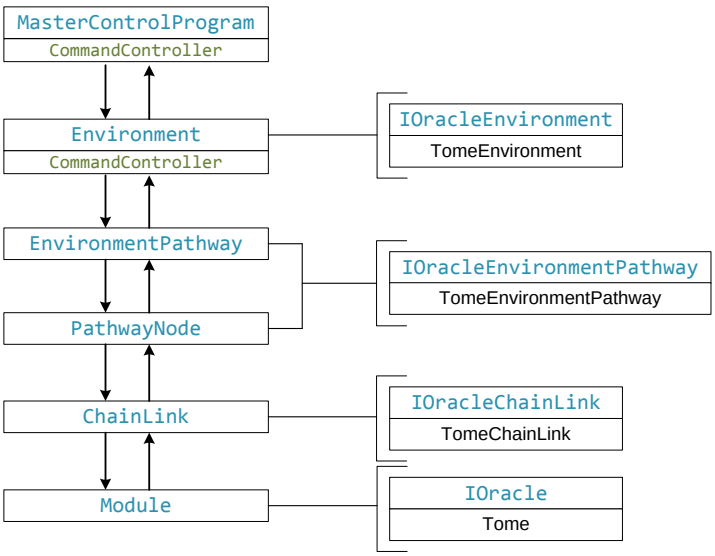


Figure 4.17 Communication hierarchy and information storage

Information stored by objects within the framework is held through a separate interface called the '*Oracle*' which has access to the '*Tome*' of knowledge each object might be holding. This acts as a separate layer and focal point for access to any framework object information that might be stored and retrieved. Figure 4.17 shows all of the 'Oracle' and 'Tome' objects within the framework in the chain of command from the highest level in the environment down to the lowest level the module object. The most important information store are those associated with the module object as they hold key information such as the *module* initialisation data, what

'input', 'output' and 'use' modules can be connected to the module along with *module* specific parameter data.

4.3.3 Framework module components

The *module* object within the framework acts as the workhorse for the entire function or process a user should wish to create. They can be seen as an equivalent to objects within object-oriented programming however with much more built in utility and structured to function as a unit within a greater whole. Looking over the structure of evolutionary algorithms, in particular genetic algorithms, the basic steps of the evolutionary process are small in number consisting mainly of selection, replacement, fitness assignment and variation operators. A module can be programmed to simply replace one of these operators, for example the 'SBX Crossover' found within the NSGAII algorithm [45] and provide the same function the original instantiation provided.

The two main objects within its scope are the *data object collection* and *module tome* objects as shown in figure 4.18. Any module can instantiate and hold any number and type of data object and allow access to any other module within the framework. The *module tome* of knowledge contains all *module* specific information that may be required to perform its function as shown in figure 4.19. This includes for example *module* parameter data, information on what inputs and outputs the module can connect to and how they are connected.

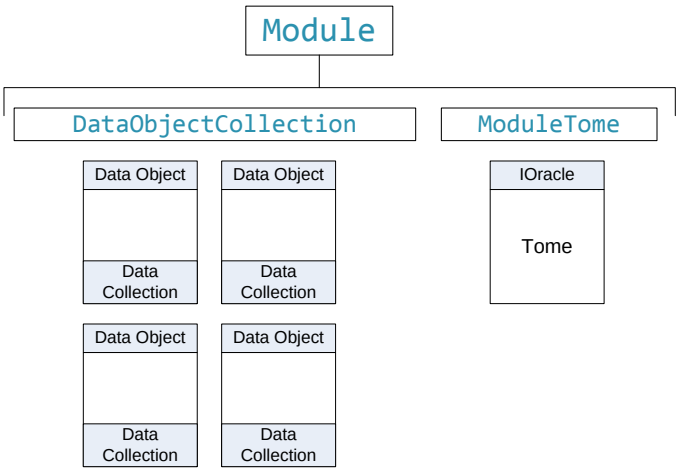


Figure 4.18 Module object scope

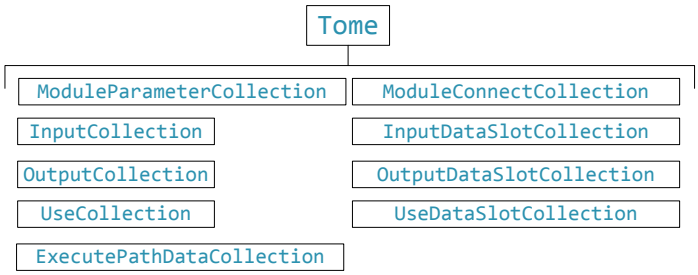


Figure 4.19 Module tome of knowledge object scope

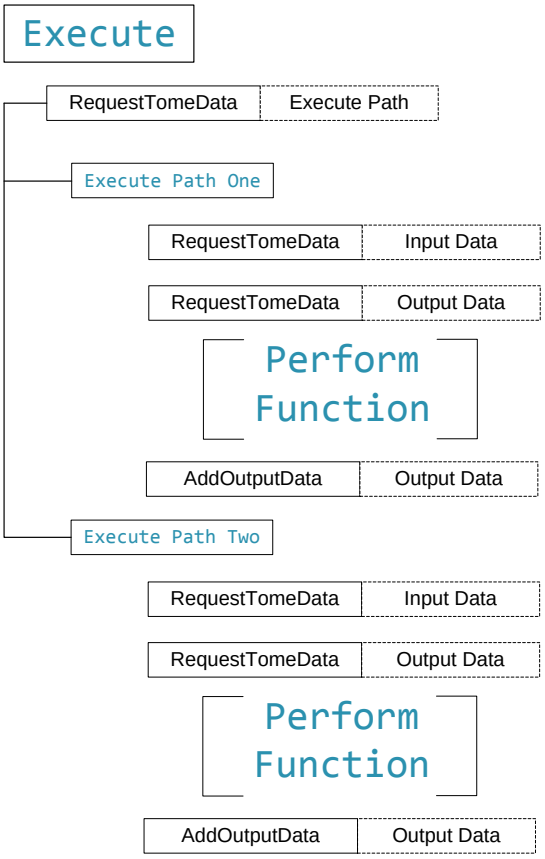


Figure 4.20 Execute function of module object

The *module* object within the framework contains for any user one section of code which provides the area of specialisation for the *module*. This code section or method called '`Execute`' shown in figure 4.20 provides the user with a template for accessing input and output data and therefore a means to alter it in some fashion. The `execute` method also contains a logic check on what `execute` path to undertake given the input and output data held within the configuration file at runtime for this *module*. Therefore different functions can be undertaken depending on what inputs or outputs are given to the *module*. A simple example of this may be when in a genetic algorithm a variation operator such as mutation looks to alter a solution or decision variable for some design problems there may not be any constraints present and therefore only input and output data is needed. However if constraints are present within the design problem and a user wishes to restrict constrained solutions it is possible for a separate execute path which contains input, output and an additional 'use' module for constraint checking to be used to force the mutation operator to start again if a solution is altered and then constrained.

Within each *module* contains a method for initialising any *module* specific parameters that might be needed for functionality. These parameters can be of any type and simply require a unique name to distinguish it as shown in figure 4.21. An example would be mutation or crossover invocation percentage parameters common to each of the operators in a typical genetic algorithm. These *module* parameters can be defined within the module in what are default values or overwritten within the configuration file or dynamically changed during the design process.

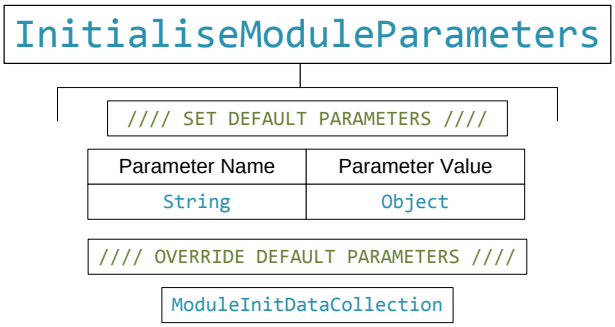


Figure 4.21 Module object parameters method

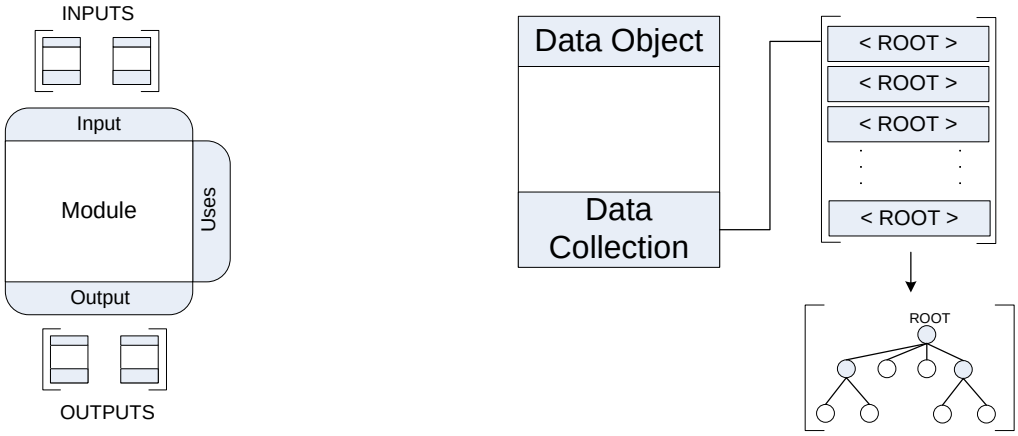


Figure 4.22 Module data object: the main unit of information

Where *modules* are used to perform some function or process with multiple connected modules providing further more complex behaviour there is still the requirement for the data it is to function on. In a typical genetic algorithm this data is often a collection of solutions each of which is often an array or vector of individual decision variables or bits. Here the *data* object shown in figure 4.22 is the unit for which modules can act upon or perform some function and are the transporters and holders of such information needed by *modules* and the overall framework process. The *data* object contains a *data collection* object itself which holds what is the core structure of all information held within the *data* object, the *root* node. The *root* node is a tree structure of *leaf* and *branch* nodes, constructed by the user or framework to hold the information. Its construction allows for any number of simple or complex tree data structures.

4.3.4 **Solution representation**

The *module* components within the framework allow for some form of function to be constructed and the structure and control of the framework allow for a designer to build complex systems or heuristics to solve or perform some kind of task. However another important part of this equation is the ability to represent and store the data that such a system is designed to act upon. As discussed previously the *data* object is the main unit of information storage within the framework and it contains a `collection` variable which holds the individual units of data. The data held within the framework *data* object is constructed from a structured tree based object consisting of *root*, *leaf* and *branch* nodes. As shown in figure 4.23 these nodes

can be used to construct complex tree based representations to store data or collections of data within each leaf node.

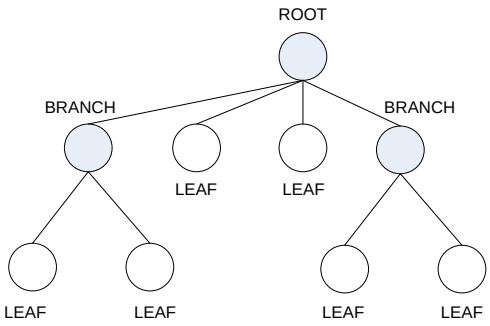


Figure 4.23 Data tree structure containing: root, leaf and branch nodes

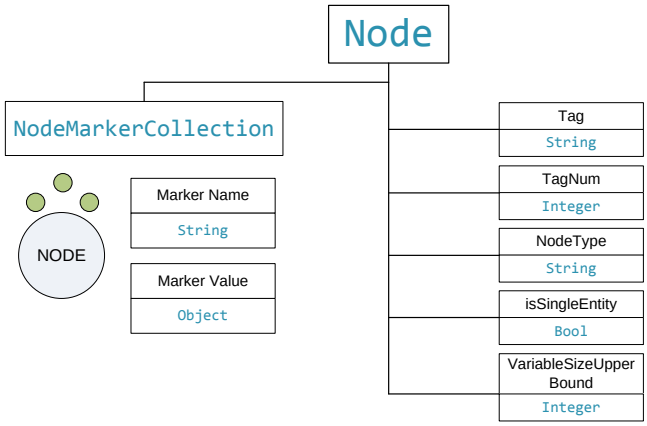


Figure 4.24 Node object scope

Each of the *root*, *branch* and *leaf* objects are derived from a lower class *node* object which holds all basic information need within the tree data structure as shown in figure 4.24. One of the most important aspects of the *node* object is the *node marker collection*. This holds *node* specific information for this particular *node* and can therefore be used to mark it with information needed for some *module* function. A simple example of its use would be in control on which of the *leaf* nodes within a *root* structure can be varied by a specific *module*. In the standard genetic algorithm the variation operators such as mutation often run through each of the decision variables of the provided solution representation, often an array of values, one by one. Here it is possible for a user to mark which variables are to be mutated and which are to be left alone, and given a mutation module which can check for these *node markers* only those chosen by the user are varied. This is just a simple example of what a system can provide, however each marker can contain more than just simple information, as the marker value can be any object provided by the C# programming language or user created class.

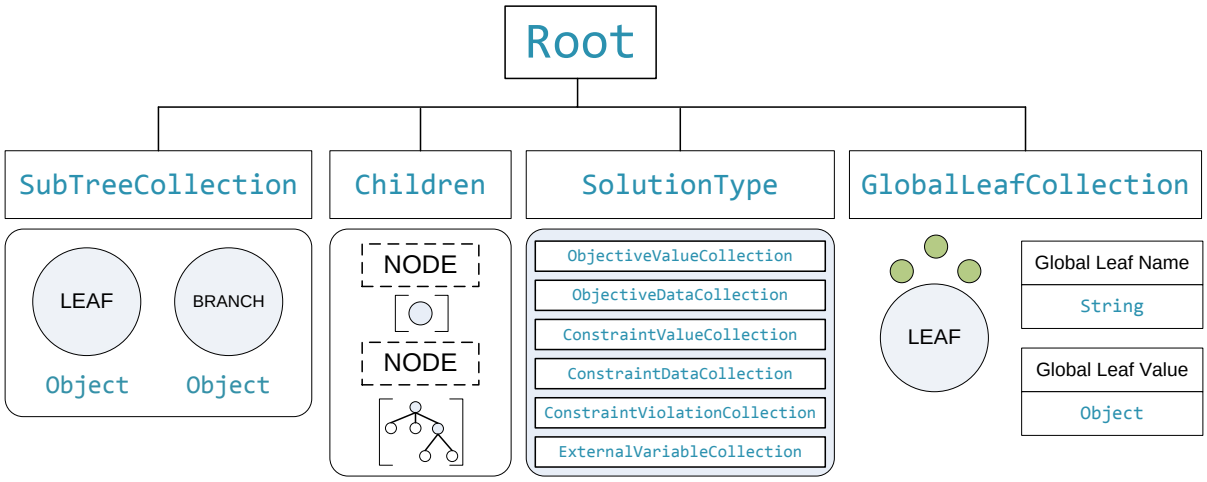


Figure 4.25 Root node object scope

The *root* object shown in figure 4.25 acts as the base of the data tree structure and first level of access for any user wishing to retrieve or alter data held within it. The main objects of interest held within it are the *sub tree collection*, which holds all tree structures which can be added to the *root* or underlying structure. It is this collection which specifies what kind of data the root object will store, a simple example would include a single *leaf* 'subtree' unit which contained a '`double`' variable. This can be used to create an array of variables to mimic a real-valued chromosome representation present in many genetic algorithms. Adding other *leaf* node variables of different types and the inclusion of more complex *branch* structures also provides a user with a wide variety of tree based representations to construct.

The *children collection* holds all children of the *root* node, while the *solution type* object is a specific set of collections for holding information for this *root* node, such as objective value data or external variables a user may wish to add to a particular piece of data. The *global leaf collection* object stores information on all *leaf* nodes within the structure that are global. Global *leaf* nodes represent a single *leaf*, whose information stored within it overwrites all other *leaf* nodes which share its type and tag name. This essentially allows for multiple *leaf* nodes to be controlled by a single master node whose variation can then be passed on to every node under its control.

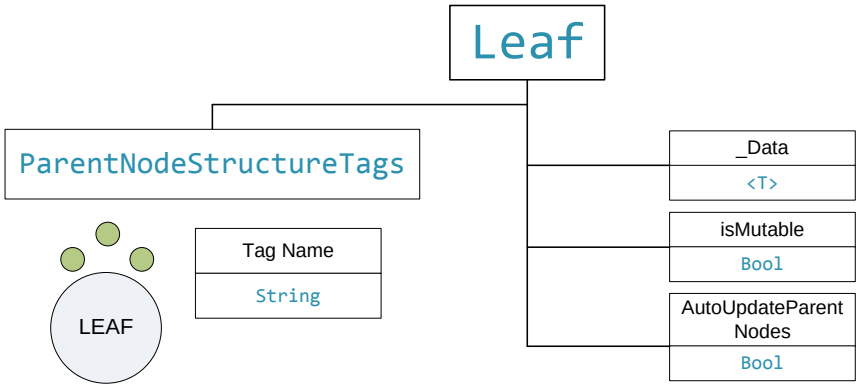


Figure 4.26 Leaf node object scope

The data within the tree based structure is held in the *leaf* nodes outlined in figure 4.26, which provide simple containers for storing and accessing data during the operation of the framework. This data can be of any type supported by the C# programming language or a user created class object so long as it is given a unique tag name. Also present is logic boolean variable '`isMutable`' to indicate whether this data can be altered, and a *parent node structure tags collection*, which contains information on whether this leaf node controls other leaf nodes. This last feature allows for example varied length structures to be constructed with specific nodes chosen to control the number of other nodes present in the tree based structure. An example would be a binary representation which contained a simple 'bit' *leaf* node and a 'bit count' *leaf* node which controlled the number of bits within the tree-based structure.

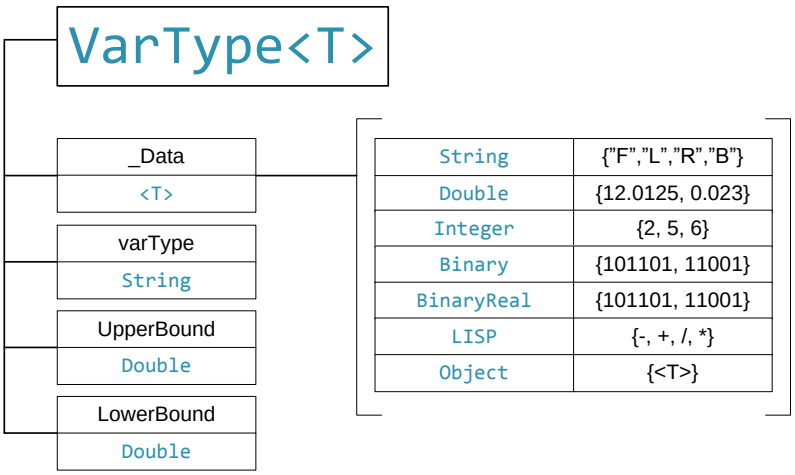


Figure 4.27 VarType object scope

Held within the *leaf* node object and shown in figure 4.27 is the *vartype* object, this is a generic object designed to hold the data or information specified by the user, and simply contains a type object to indicate what is held within it and a set of upper and lower bound objects if needed to control the data range.

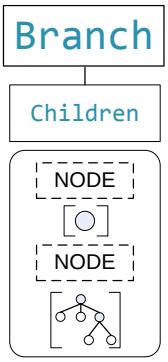


Figure 4.28 Branch node object scope

The final unit within the data representation and shown in figure 4.28 is the *branch* node which simply acts as a method for adding structure within the whole tree data object and therefore contains simply a collection of all children associated with it.

4.4 Multi-objective evolutionary algorithm validation

In order to undertake the experimentation within this thesis the necessary algorithms will have to be constructed using the framework outlined previously. Chapter 2 outlined the recent advances in automated MEMS design optimisation with a particular focus upon the field of evolutionary computation. A number of experiments were undertaken using the stochastic multi-objective genetic algorithms NSGAII [45] and MOGAII [44] in order to establish their efficacy and robustness in MEMS design. The success of these multi-objective genetic algorithms towards automated design optimisation therefore highlights the need to build such algorithms within the framework outlined. Within the literature two of the most common and state of the art multi-objective genetic algorithms are NSGAII used previously and SPEA2 [132] a separate algorithm designed by Zitzler *et al*.

The multi-level strategies outlined here and the standard single level strategies they are tasked to improve upon both rely on the multi-objective genetic algorithms to perform. The choice of NSGAII and SPEA2 allows a wider range of optimization to be tested rather than spending more time on a single MOGA which may have diminished performance for some design problems. The use of two algorithms also provides a more robust analysis into the performance of both the multi and single level strategies for MEMS design optimisation.

Before each algorithm can be used to evaluate the role multi-level design optimisation can have in MEMS design they need to be validate against a common design problem within the literature and also compared against another commonly used framework to ensure the results achieved are similar. One of the most used alternative multi-objective genetic algorithm frameworks is jMetal [270] a java-based evolutionary computation platform that has over 75 publications to its name. The chosen design problem to test each of the algorithms is the classic 'Golinski' or speed reducer problem used to test a number of multi-objective and multidisciplinary optimisation algorithms within the literature [254][255][257][258][261].

The 'Golinski' problem looks to optimize the sizing of a speed reducer component and originally formulated as a single objective problem [271] it has also been expanded into a two [254] and three objective [255] design problem with formulations for multidisciplinary optimisation also constructed within [261]. The speed reducer in figure 4.29 consists of 7 design variables all tied to the component with the objectives set out to minimize the volume while simultaneously reducing the stress placed upon the shafts.

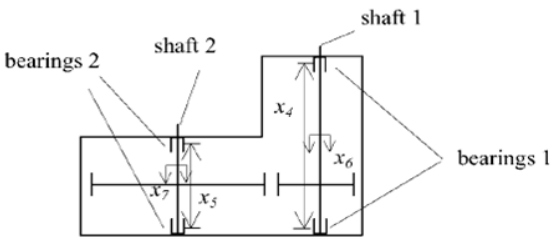


Figure 4.29 Speed reducer problem model taken from [azarm]

The objectives for the design problem are shown in equation 4.1 for both the two and three objective problem. In the case of the two objective design problem only objectives f_1 and f_2 are

used. Also associated with the speed reducer problem are 11 inequality constraints outlined in equation 4.2.

$$f_1 = 0.7854\, x_1 {x_2}^2 \left(\frac{10{x_3}^2}{3} + 14.933\, x_3 - 43.0934\right) - 1.508\, x_1({x_6}^2 + {x_7}^2) + 7.477({x_6}^3 + {x_7}^3) + 0.7854(x_4 {x_6}^2 + x_5 {x_7}^2)$$

$$f_2 = \frac{\sqrt{\left(\frac{745\, x_4}{x_2 x_3}\right)^2 + 1.69 \times 10^7}}{0.1{x_6}^3} \qquad f_3 = \frac{\sqrt{\left(\frac{745\, x_5}{x_2 x_3}\right)^2 + 1.575 \times 10^7}}{0.1{x_7}^3} \tag{4.1}$$

$$g_1 \equiv \frac{1}{x_1 {x_2}^2 x_3} - \frac{1}{27} \leq 0; \quad g_2 \equiv \frac{1}{x_1 {x_2}^2 {x_3}^2} - \frac{1}{397.5} \leq 0;$$

$$g_3 \equiv \frac{{x_4}^3}{x_2 x_3 {x_6}^4} - \frac{1}{1.93} \leq 0; \quad g_4 \equiv \frac{{x_5}^3}{x_2 x_3 {x_7}^4} - \frac{1}{1.93} \leq 0;$$

$$g_5 \equiv x_2 x_3 - 40 \leq 0; \quad g_6 \equiv \frac{x_1}{x_2} - 12 \leq 0;$$

$$g_7 \equiv 5 - \frac{x_1}{x_2} \leq 0; \quad g_8 \equiv 1.9 - x_4 + 1.5 x_6 \leq 0; \tag{4.2}$$

$$g_9 \equiv 1.9 - x_5 + 1.1 x_7; \quad g_{10} \equiv f_2 - 1800 \leq 0;$$

$$g_{11} \equiv \frac{\sqrt{\left(\frac{745\, x_5}{x_2 x_3}\right)^2 + 1.575 \times 10^7}}{0.1{x_7}^3} - 1100 \leq 0;$$

The next task is to build the correct modules and representations in order to solve the speed reducer problem. Table 4.2 holds the decision variables for the speed reducer problem along with their type and upper / lower bounds. The representation that will store this information needs to be able to handle mixed variable types of real-valued and integer values. Therefore a new representation has been included within the framework, coined '`MixedIntegerRealValuedChromosome`', which contains the sub trees for real-valued and integer data types as shown in figure 4.30. This data template can then be used to create a mixed value chromosome encoded for the speed reducer problem, first through creation of a seven leaf structure containing both integer and real-valued data types as seen in figure 4.31 and then applying upper and lower boundary values and later initialisation as shown in figure 4.32. All this is achieved through the creation of a problem specific configuration file.

Table 4.2 Speed Reducer Variable Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Variable 1	Real-Valued	2.6	3.6
Variable 2	Real-Valued	0.7	0.8
Variable 3	Integer	17	28
Variable 4	Real-Valued	7.3	8.3
Variable 5	Real-Valued	7.3	8.3
Variable 6	Real-Valued	2.9	3.9
Variable 7	Real-Valued	5.0	5.5

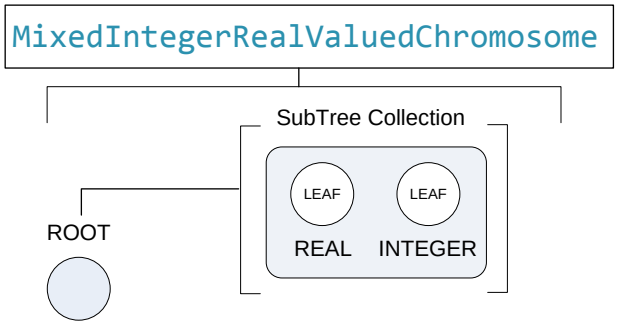


Figure 4.30 Mixed integer and real-valued chromosome data template

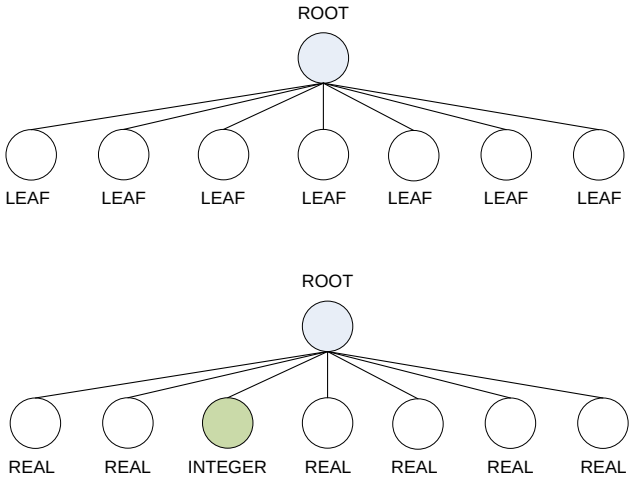


Figure 4.31 Tree based representation: from node type to value type

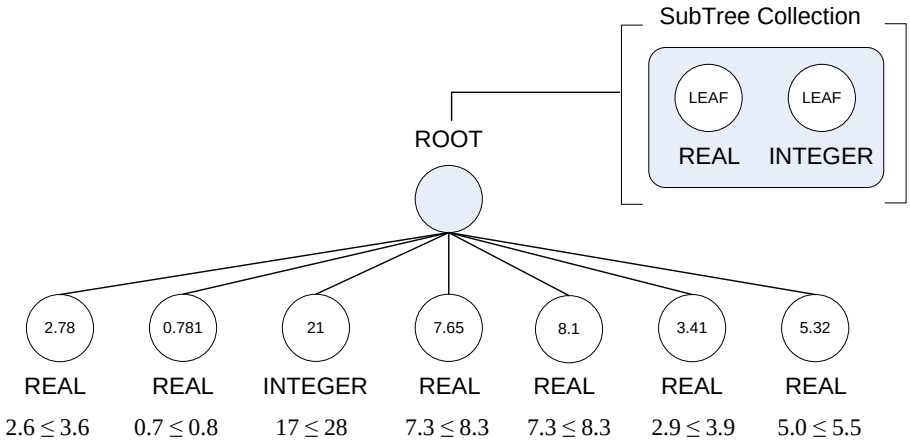


Figure 4.32 Root tree structure for a solution to the speed reducer problem, indicating: data type, data value and upper and lower bounds

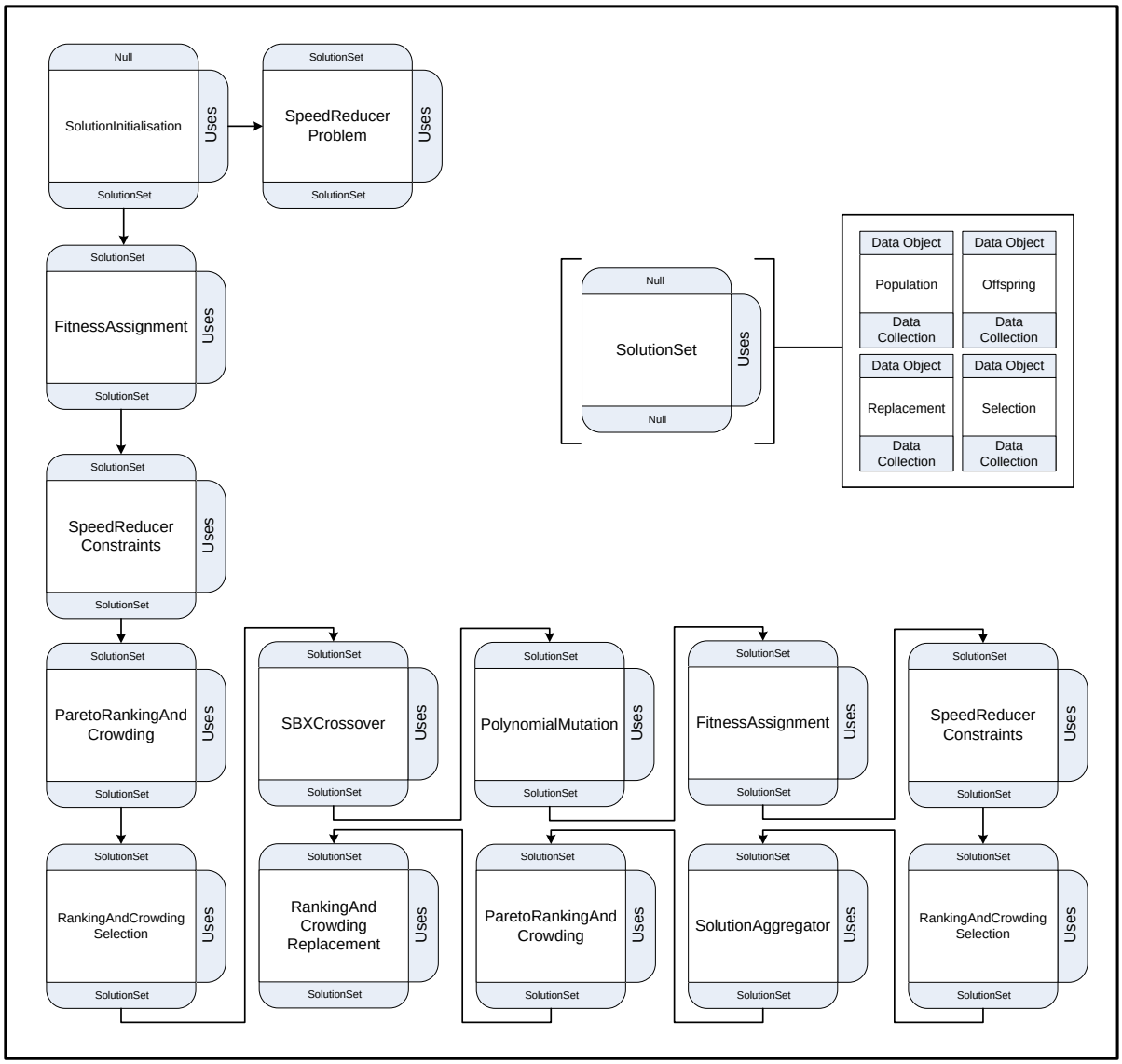


Figure 4.33 Default speed reducer design problem pathway for NSGAII

The realisation of the representation to hold the decision variables associated with the design problems having been completed the next stage is construction of the entire design process for each of the multi-objective genetic algorithms chosen to solve this particular problem. The current state of the art NSGAII and SPEA2 algorithms have been deconstructed into their unassembled components and then formulated into individual modules within the framework and reconstructed into an environment pathway to solve the design problem as shown in figure 4.33.

The NSGAII algorithm environment pathway contains all the operators and heuristics associated with the original incarnation described by Deb [45] and outlined in appendix A. Here the environmental pathway starts as most standard algorithms with an initialisation of the starting population where four modules play an important role. To begin the solution initialisation module ‘uses’ the separate speed reducer problem module to gather the information it requires to build the population. This information contains representation and variable data along with

objective and constraints associated with the design problem. The solution initialisation module works on user specified data objects, in this case the population data object set which itself is retrieved from a solution set module. This module simply acts as a retainer of the data object solution sets that will be used throughout the design process and contains solutions sets typical to the NSGAII algorithm.

After the population set has been randomly initialised each of the solutions within the population data set has to be evaluated using a fitness assignment module. Because the objective information contains all the necessary data and equations to work this out, all active objectives are evaluated within the module. Finally each of the solutions is evaluated for constraint violation using the speed reducer constraints module and as with the fitness assignment module this is undertaken internally for each solution. These four modules of the environment pathway complete what is the standard initialisation phase of most genetic algorithms and as such are only active for a single cycle, and therefore their individual pathway node logic has a run count of one and a run step count of zero.

The next stage of the pathway handles the rest of the design process from start to finish and contains the operators tasked with the main processes of selection, variation and replacement. Each cycle occurs in a linear fashion starting with the Pareto ranking and crowding module and finishing with the ranking and crowding replacement module the population is evolved to solve the design problem at hand. The design process is stopped when the number of designated functional evaluations has occurred and the run expression of the pathway logic is evaluated as false halting the entire process.

A similar pathway has been constructed for the SPEA2 algorithm and both are now to be validated on the two and three objective speed reducer problem outlined previously. But first there is another algorithm that needs to be constructed and tested as well, for even though the multi-level strategies all employ the basic NSGAII or SPEA2 pathway set out, the multidisciplinary optimisation algorithm itself contains a more complicated structure and is as yet untested.

4.4.1 Multidisciplinary optimisation problem formulation

In applying the MDO algorithm we first begin with the decomposition of the design problem into a number of subsystems each with their own decision variables, local objectives and constraints. The decision on how this decomposition is undertaken is up to the user and within the MDO literature there are a number of methodologies two of which, aspect and object have been discussed. There are also similar methods for identifying the important functions, analysis and objectives in a design problem for example axiomatic design [265] which can also lend their support.

The decomposition of a multi-objective problem into a number of subsystems is shown in figure 4.34. Here the default design problem is held and optimised within the system level with the original objectives f_{01} / f_{02} and constraints g_0 active and a chosen set of decision variables X_{sh} open for variation. The decision on what variables are included within the X_{sh} set are up to the designer however they are often decision variables that are common to more than one subsystem [261] and often hard to separate so are shared throughout all subsystems. All other decision variables are closed to the system level and remain fixed.

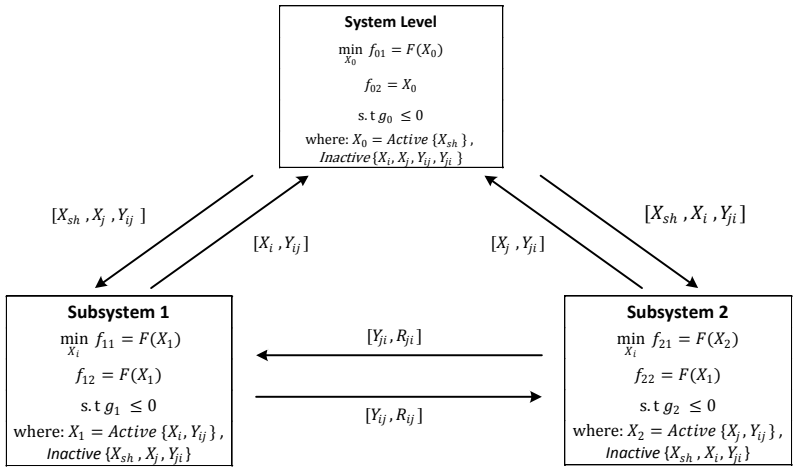


Figure 4.34 Multidisciplinary optimisation non-hierarchical structures for decomposed problem

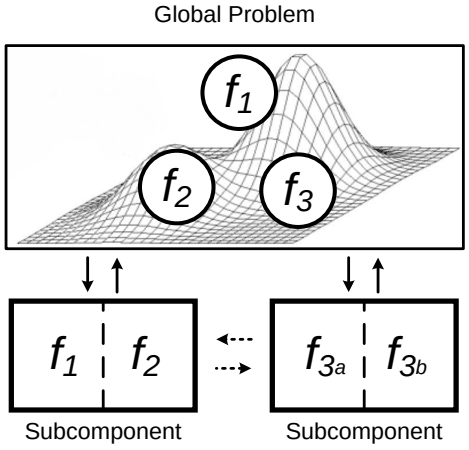


Figure 4.35 Fitness assignment: three objective global engineering problem and two differing subcomponent fitness assignment strategies

The subsystems are constructed as a non-hierarchical design with communication both from the system to subsystem or parent to child level and from subsystem to subsystem occurring. Each of the subsystems contains its own local objectives f_{11} / f_{12} and these can be unique, additively separable from one of the system level objectives or one of the system level objectives in its own right as shown in figure 4.35. In a similar vein the constraints g_1 / g_2 held within each subsystem can also be unique or taken from the system level design problem. The active decision variables within each subsystem consist of local disciplinary design variables X_i / X_j and the coupled disciplinary design variables Y_i / Y_j. Where the local disciplinary design variables are fixed to each subsystem the coupled design variables are not and as a result they are transferred from their local subsystem to all other subsystems within the structure every cycle. Finally not all problems can be fully decomposable in respect of their disciplinary analysis and as a result more than one system may rely on information garnered from another. Therefore when applicable coupled analysis response variables can also be passed between the child subsystems, with the origin of the subsystem analysis passing on these variables to any other subsystem that requires them. A default chromosomal representation of the various design and response variable sets for a single solution is shown in figure 4.36.

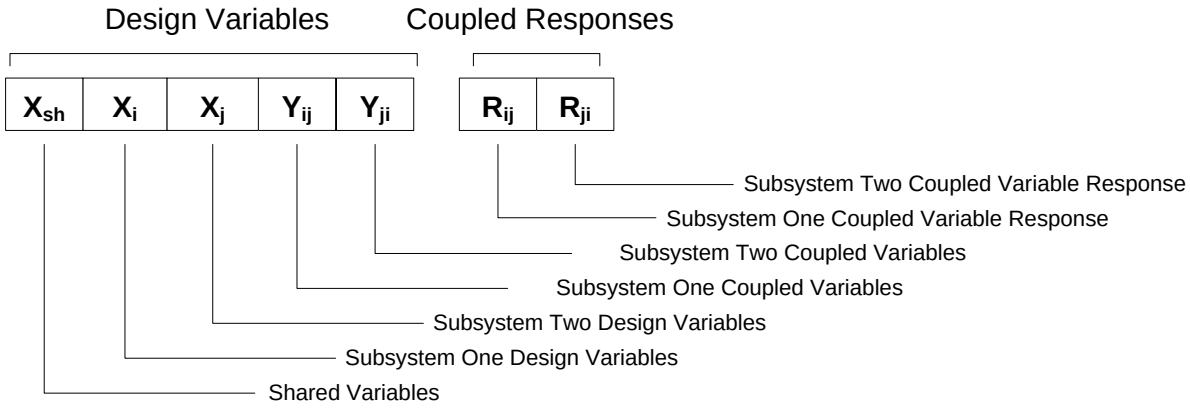
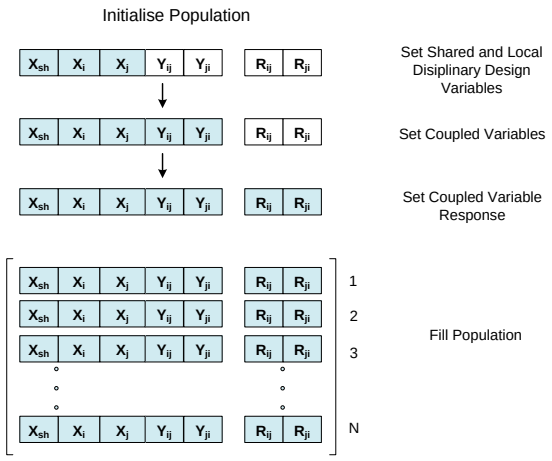


Figure 4.36 Multidisciplinary optimisation design and response variable sets for a chromosomal representation

The overall process of the multidisciplinary optimisation algorithm can be broken down into a number of key steps, in this instance linked to a multi-objective population based optimizer and they are described below.



Step 1. The first step begins with the initialisation of the system level population; in particular the various variable sets as shown in figure 4.37. Any coupled variable response values are set to null to be filled later after functional evaluation. The filled system level population set 'popCurrent' is than ready for variation. Functional evaluation of each individual is based upon system level objectives $\{f_{01}, f_{02}\}$

Figure 4.37 Population initialisation for system Level

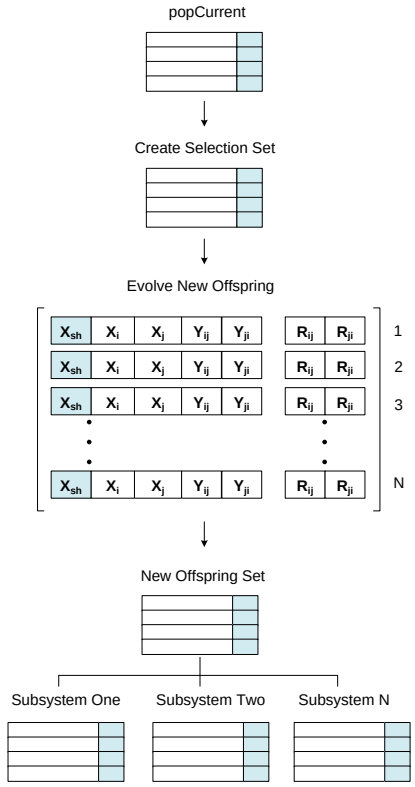


Figure 4.38 System level to subsystem level transition

Step 2. A selection set is chosen from the current system level population ready for variation and the creation of an offspring population set. Only the system levels 'shared variables' X_{sh} are varied based upon the chosen optimizers' operators. At the system level this offspring set is then used depending upon the chosen the optimizers' replacement operators as the basis for the next popCurrent set. However as shown in figure 4.38 the newly created system level offspring population set is also passed on to the each of the subsystems within the multidisciplinary optimisation structure.

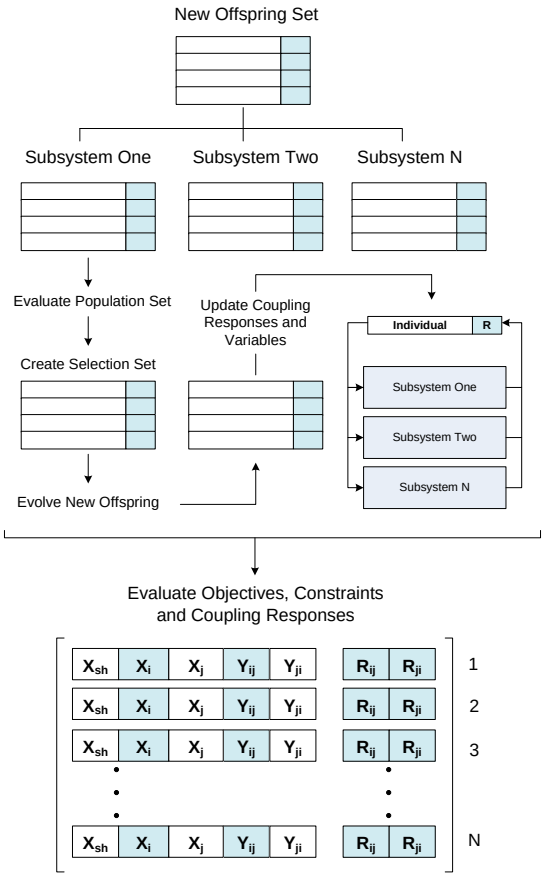


Figure 4.39 Subsystem level design process

Step 3. The next step shown in figure 4.39 moves on to the subsystem level of the design process, upon receiving the offspring sets the individual solutions are used to fill the local subsystem populations. Subsystem populations with a lower number of solutions than the supplied offspring set are filled using a truncation operator. For each subsystem the local population sets now need to be evaluated using local objectives $\{f_{11}, f_{12} / f_{21}, f_{22}\}$ and constraints before then undertaking a standard routine of selection, variation and replacement. Each of the subsystems variation operators are restrained to only alter their local disciplinary design variables X_i / X_j and the local coupled disciplinary design variables Y_i / Y_j. After variation has occurred, any coupled variable within each subsystem offspring solution is passed on to all other subsystems, as a result all subsystem offspring set sizes are fixed to the same size. Finally functional and constraint evaluation of each subsystem offspring population set is undertaken and where necessary coupled disciplinary analysis variable values are also transferred to any subsystem solutions that may require them for functional or constraint evaluation. The local subsystem offspring sets are then combined with their local population sets before replacement operators update each subsystem with a new population set. This iterative process then continues for a fixed number of cycles before ending and moving on to the next step.

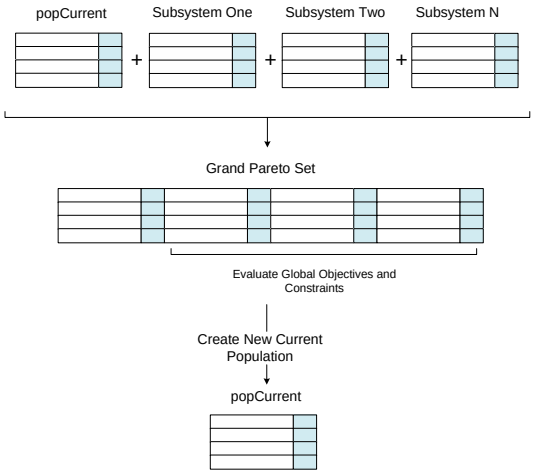


Figure 4.40 System level popCurrent population set update

Step 4. The final step in figure 4.40 looks to take the evolved subsystem population sets and combine them into a 'Total Population' set for evaluation of objectives and constraints at the system level $\{f_{01}, f_{02}\}$. The size of the total population set is fixed to the total sum of all subsystem population sets. This total population set is then combined with the system level popCurrent set to form a unified 'Grand Pareto' set which is then used to create a new popCurrent set using the optimizers' replacement operators. Upon completion of this step the process begins again at step two and the whole process is repeated until a chosen criterion is used to determine whether it should be stopped

One of the biggest changes is in the decomposition of the design problem into a number of tasks or subsystems and as a result to break up of the representation that encodes the problem. The multi-objective population based multidisciplinary optimisation algorithm for solving both the two and three objective speed reducer problem uses the same decomposition methodology set out in [254] for the two objective and [255] for the three objective problem. Each of the methodologies involves decomposing the decision variables into two separate subsystems and focusing each subsystem on different objectives linked to the original problem found at the system level. Figure 4.41 shows the representations used for the default system and decomposed subsystem levels with the active decision variables and associated node markers. The makeup within the framework of the representations used to store data provides useful tools to allow easy construction of complex structural algorithms like multidisciplinary optimisation. Here each leaf is given a node marker to indicate which is active for that particular system or subsystem and the modules within it, in particular variation operators such as mutation and crossover. The environment pathway for the multidisciplinary optimisation algorithm is similar to figure 4.33 except for each system and subsystem level it contains a specific NSGAII algorithm. There are also additional modules included within the pathway specific to MDO, with a solution duplicator module used to pass on copies of the system level offspring to each of the subsystems and a gene swapper module for passing the coupled decision variables between each subsystem offspring all in accordance with the process outlined above.

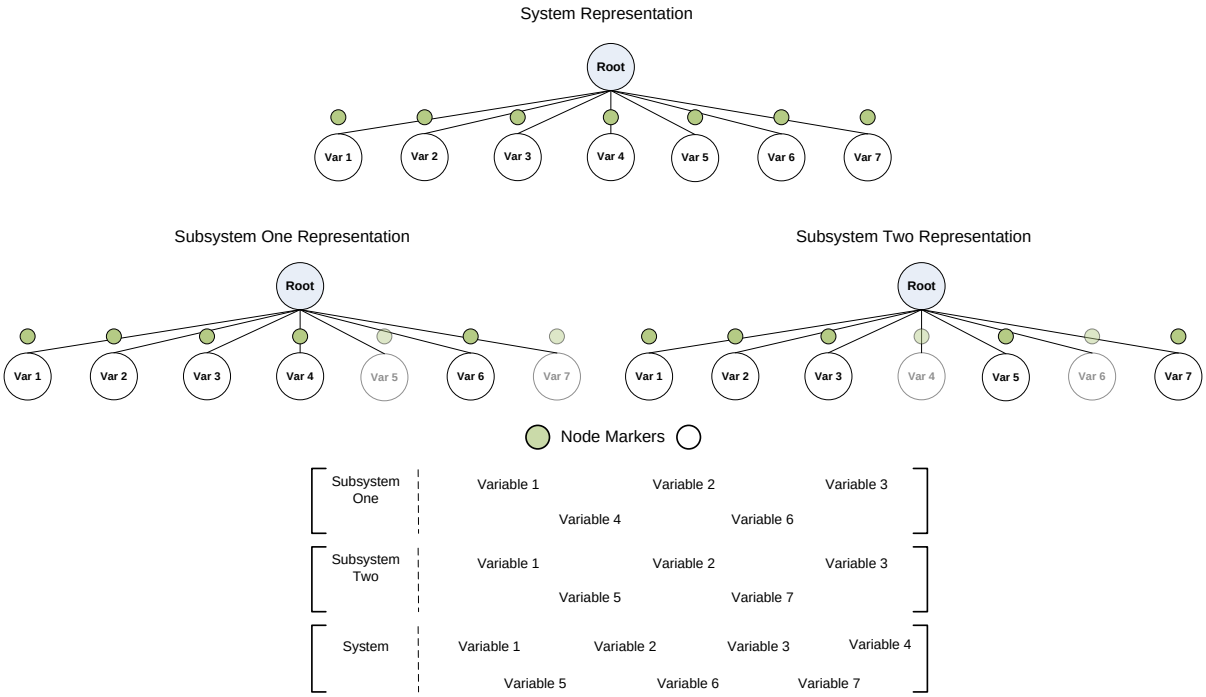


Figure 4.41 Multidisciplinary optimisation representations and associated node markers for speed reducer two and three objective design problem.

With all three algorithms built within the framework the task to validate them can be undertaken. Both the two and three objective speed reducer design problems are used and for each algorithm a set of five tests each consisting of 10,000 functional evaluations have been undertaken. The default algorithmic parameters for both NSGAII and SPEA2 present within the modules of each environment pathway for this problem are shown in table 4.3 along with the various population sizes. The multidisciplinary optimisation algorithm contains a hierarchical structure which utilises a number of additional populations sets, the default values for the size of each of these sets is also shown in table 4.4.

Table 4.3 Default Algorithm Parameters NSGAII and SPEA2

Algorithm Parameter	Default Value
Population Size	100
Offspring Size	100
Selection Size	100
Replacement Size	200
SBX Distribution Index	20
Polynomial Mutation Distribution Index	20
Probability of SBX Crossover	0.8
Probability of Mutation	0.142857
Generations	100
Tests	5

Table 4.4 Population Parameters Multidisciplinary Optimisation

Algorithm Parameter	Default Value
System Population Size	100
System Offspring Size	100
System Selection Size	100
System Replacement Size	200
Grand Pareto Size	300
Subsystem Population Size	100
Subsystem Offspring Size	100
Subsystem Selection Size	100
Subsystem Replacement Size	100
Subsystem Total Size	200

The design process for both the NSGAII and SPEA2 algorithms are linear and structurally very simple, looping every cycle from start to finish, however the multidisciplinary optimisation algorithm interchanges between system and subsystem levels periodically. In the MDO design process the system level and its associated NSGAII algorithm is called every ten cycles and involves updating the total population and grand population sets with the evolved subsystem population sets. The subsystem level and their associated NSGAII algorithms are called sequentially every cycle and as a result around 1000 functional evaluations are used at the system level and 9000 at the subsystem level.

4.4.2 Speed reducer results and analysis

In order to assess the performance of each of the algorithms on the two speed reducer design problems the hypervolume metric is used to both evaluate the Pareto spread and dominance of the objective space each of the tests final populations has produced. The mean and bound hypervolume results for each algorithm are shown in table 4.5 with the chosen nadir point for the two and three objective problem indicated below. Also included are the results for similar experiments using a separate framework jMetal [270] for both the NSGAII and SPEA2 algorithms as a means of comparison in performance.

Table 4.5 Speed Reducer Two and Three Objective Design Problem Hypervolume Metric Values

Speed Reducer – 2 Objectives					
	NSGAII	NSGAII jMetal	SPEA2	SPEA2 jMetal	MDO NSGAII
S^U	1927425.898	1928357.573	1926371.720	1926564.794	1926094.169
S^M	1927096.298	1928174.165	1925150.499	1924341.543	1925560.005
S^L	1926533.553	1927877.019	1921631.936	1920820.293	1924012.468
Speed Reducer – 3 Objectives					
	NSGAII	NSGAII jMetal	SPEA2	SPEA2 jMetal	MDO NSGAII
S^U	669348872.86	668094643.12	674722574.432	674407595.97	672953351.4
S^M	667099790.88	665813515.12	667656079.534	671818559.07	670425237.7
S^L	664741675.34	661718263.77	660545560.310	669481314.24	666497637.0

$*(S^U S^M S^L)^1$ [5800, 1350] $*(S^U S^M S^L)^2$ [6000, 1350, 1100]

Looking at the results there is variation between the modular framework and jMetal results with NSGAII slightly underperforming against the NSGAII jMetal version and with SPEA2 performing slightly better against the SPEA2 jMetal version for the two objective problem. This trend is reversed for the three objective problem, however overall the discrepancy between both the

modular framework and jMetal is small, something backed up by the final Pareto fronts found by both algorithms in figure 4.42.

The application of the multidisciplinary optimisation algorithm to each of the speed reducer design problems was also successful though with some reduction in hypervolume performance for the two objective problem when compared with the default NSGAII results. Comparing the performance qualitatively with previous results of some of the more state of the art MDO algorithms are presented in figure 4.43 for the two objective and figure 4.44 for the three objective design problem showing similar or improved performance.

The two objective speed reducer comparison is taken from [254] and the application of a multidisciplinary optimisation immune network simulation based algorithm. This particular approach took 21931 functional evaluations to generate 7 Pareto points using a population size of 50 solutions in the system subsystem population sets. There are a number of other differences unrelated to the structure of the MDO used, for example a binary representation and the use of one-point crossover operators, however the use of the more state of the art NSGAII and a larger population has benefited the multidisciplinary optimisation strategy outlined to give overall superior results in terms of Pareto spread.

Looking at the three objective comparison between work undertaken in [255] using a quality-assisted multi-objective multidisciplinary optimisation genetic algorithm in figure 4.44 both the current state of the art algorithm [255] and the one outlined here show good agreement with regards to the final Pareto population set.

4.5 Summary

The need for a robust computational framework to house the multi-level and multidisciplinary optimisation algorithms and interface with the numerous modelling and analysis tools within MEMS design is also of great importance in this work. This chapter has successfully outlined, built and validated a modular framework tailored towards automated design optimisation of MEMS through two common multidisciplinary design problems. The multi-objective evolutionary algorithms NSGAII and SPEA2 were compared with a current state of the art computational framework jMetal and found to have results in good agreement. The proposed multidisciplinary optimisation algorithms outlined here were are also constructed within the framework and validated using the same set of design problems and compared against similar state of the art MDO examples found in the literature. The next chapter moves to build a hierarchical MEMS bandpass case study using the framework outlined. This case study covers a number of the hierarchical levels of modelling and analysis present within MEMS design, and is used throughout the remainder of the thesis for evaluation and comparison of the current state of the art multi-objective evolutionary algorithms (NSGAII and SPEA2) described here and the multi-level and multidisciplinary optimisation strategies from 3 and 4.

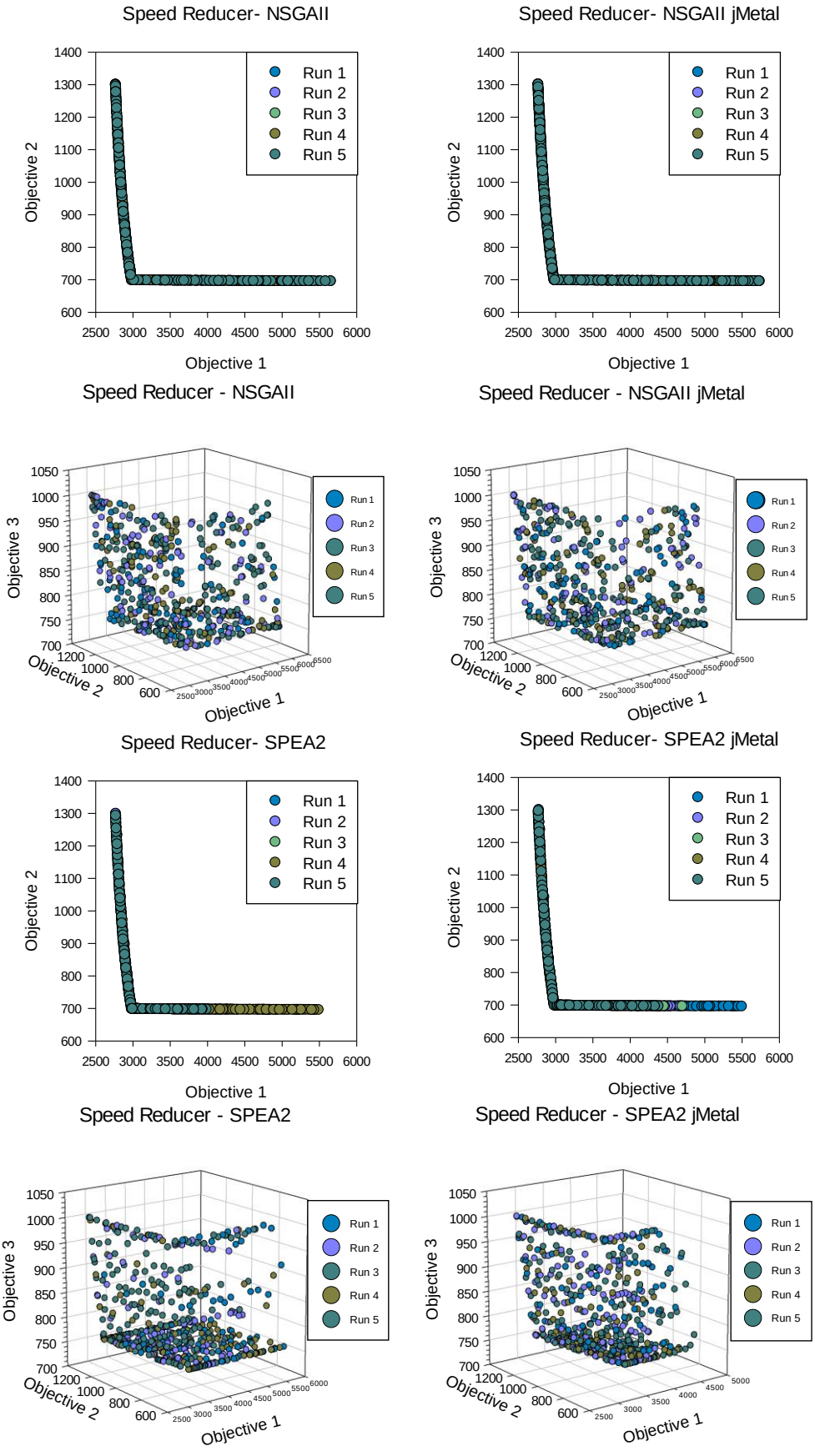


Figure 4.42 Final population sets two and three objective speed reducer problem.

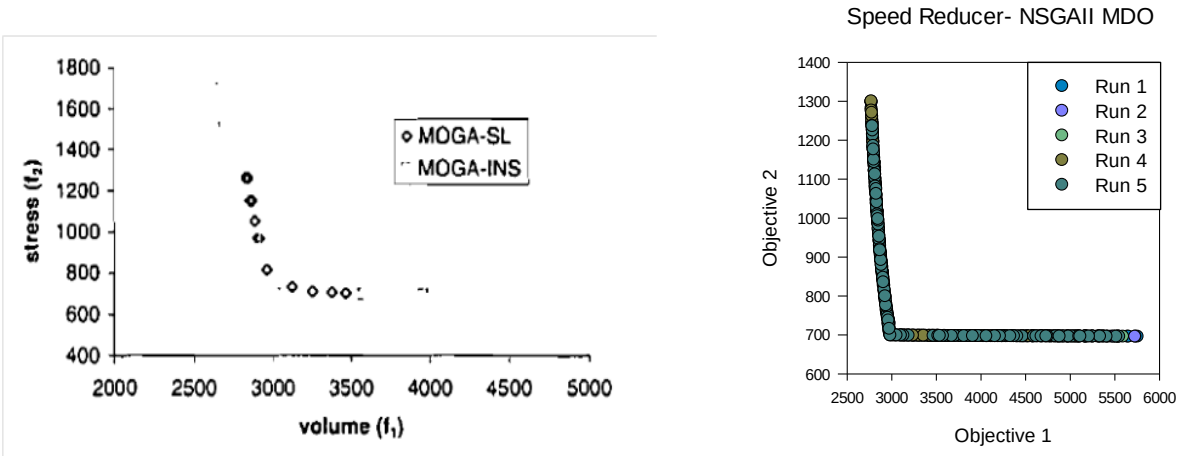


Figure 4.43 Speed reducer two objective results taken from [254].

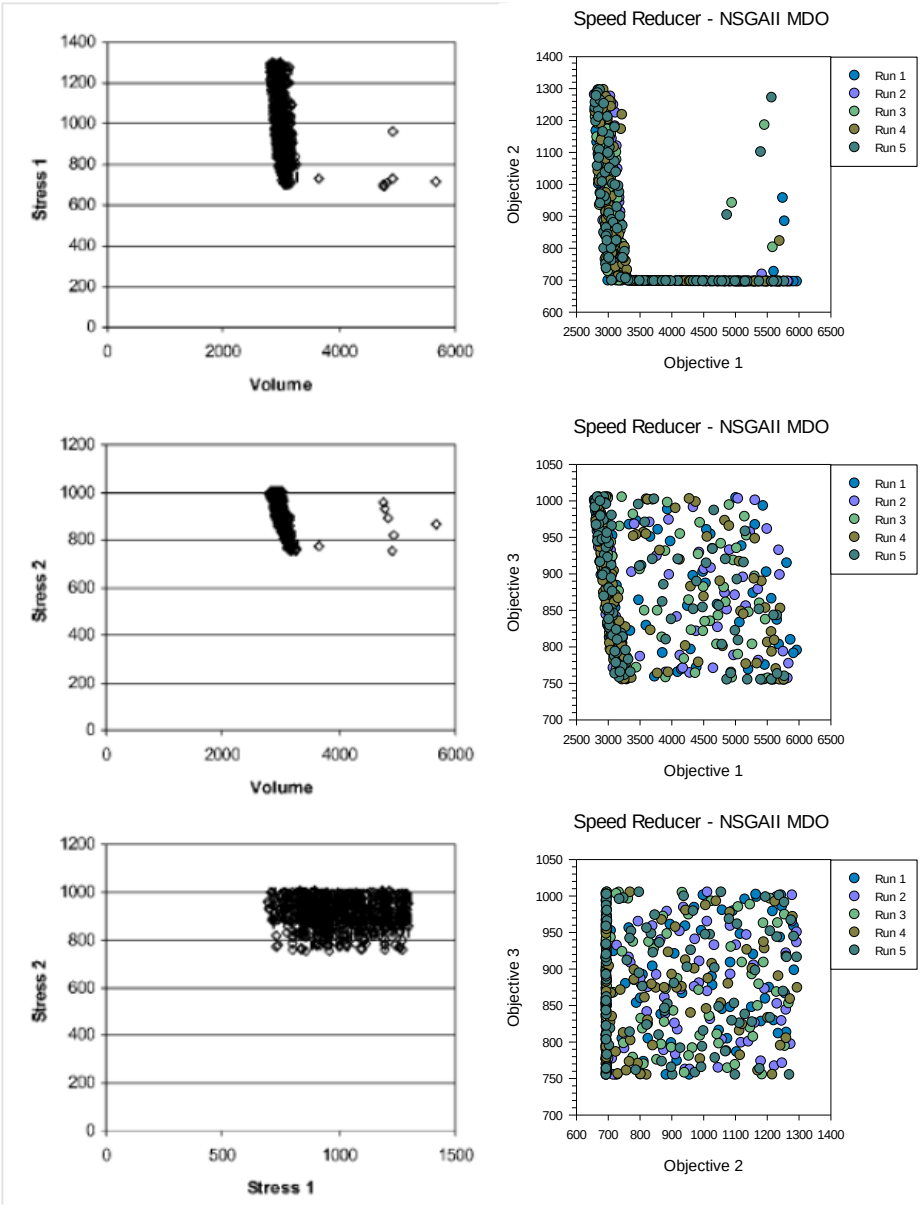


Figure 4.44 Speed reducer three objective results taken from [255].

5

Bandpass Filter Case Study

This chapter presents a MEMS bandpass filter case study to be used as a template throughout the remainder of this thesis' exploration into multi-level and multidisciplinary design optimisation of MEMS. An outline of the industrial implications of micromechanical filter design is given first; ending with the construction of electrical and mechanical modelling of a MEMS bandpass filter system. The final section concerns itself with the design synthesis and optimisation of both system and device level bandpass filter design problems in order to validate the case study as suitable for further investigation.

5.1 Proposed modelling on MEMS bandpass filter synthesis

Microelectromechanical systems offer a range of benefits over their contemporary macro alternatives and more and more are giving rise to replacing them. Often more robust and environmentally tolerant [16], they also benefit from their scale by taking up less space and requiring less power to function.

Heterodyning communications are one area where the application of MEMS could provide profound benefits [272]. A component of such communication devices are bandpass filters, particularly important in highly selective HF (high-frequency), IF (intermediate-frequency), or RF (radio-frequency) signal processing and mixing [273]. The performance characteristics of such filters such as having a low insertion loss, small percent bandwidth and shape factor are heavily influenced by the *Quality* (*Q*) factor of the component which itself must be tolerant also to environmental variations such as temperature, noise or microphonics that effect frequency transmission [272][274]. Electronic filters, such as transistor based *LC* circuits can be employed to function as filtering components; however they are limited due to their relatively low *Q* factor values [275].

The majority of heterodyne transceivers utilize macro vibrating mechanical tank components like crystal or surface acoustic wave (SAW) resonators [274][131] to perform signal processing. These particular components have advantages over transistor based technologies in comparable filters due in part to their high-*Q* factor, giving rise to superior performance in insertion loss, percentage bandwidth, and achievable rejection, and also their stability against thermal ageing [274][276][277][278].

The mechanical components outlined however do have a particular disadvantage in that they are off-chip and therefore must interface with processing circuitry at the board level, increasing total device area. This is problematic with regards to miniaturization of the system and therefore portability of wireless transceivers which has led to research on new strategies for miniaturization of these components [274][279][272].

CMOS micromachining technology allows for the fabrication of on-chip components that can interface directly with electronic interface circuitry and therefore reduce the overall device footprint. This technology has been employed in the synthesis of High-*Q* micro mechanical oscillators/resonators [131] which match target performance and cost goals designers aim for.

Some of the components which can be targeted for replacement with MEMS in wireless communications include RF filters; including image rejection filters with centre frequency ranges of 800 MHz to 2.5 GHZ; IF filters, with centre frequencies of 455 kHz to 254 MHz, and high-*Q* tuneable, low phase noise oscillators [272].

Others include MEMS varactor-based circuits [280], MEMS switch-based circuits [281], micromachined cavity resonator-based circuits [282], micromachined transmission line-based circuits [283].

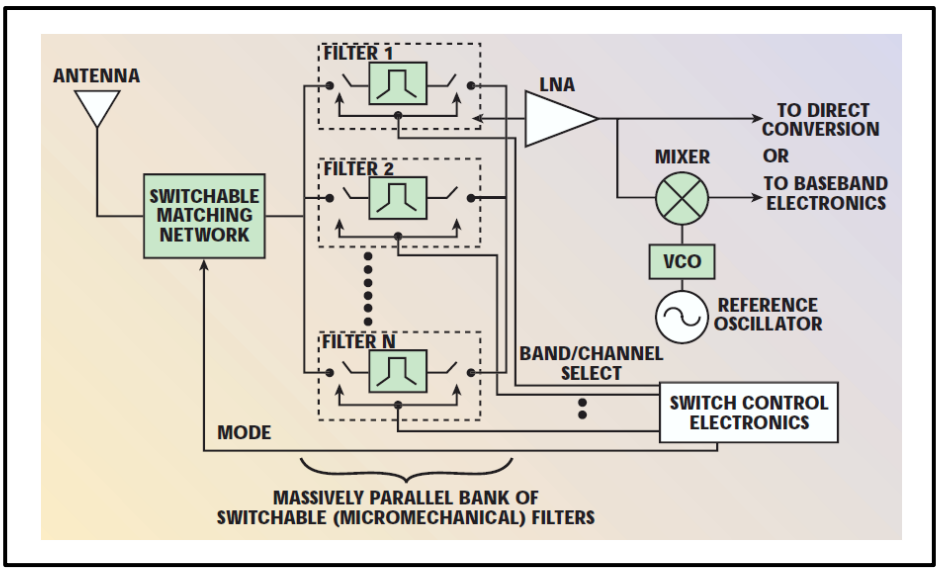


Figure 5.1 MEMS switchable receiver front-end architecture [284]

Broadband receiving and transmitting systems often require the application of tunable or wide-band filters, often realised with switching between multiple fixed-tuned circuits over a given frequency range [285].

Tuneable RF filters that utilize single High-Q resonators are difficult to attain due to the high stability they enjoy which makes voltage-controllable frequency tuning difficult over large frequency ranges. MEMS based filters have reasonable tuning ranges of around 5% from design central frequency though this is still lacking for the tuning ranges needed in applications such as wide-band filters [286].

However an alternative approach which utilizes the low power consumption and miniaturization of micromechanical filters can provide tuneable RF filters for application in broadband receiving and transmitter systems. The size and power consumption of a micromechanical filter means it is possible to fit hundreds or perhaps thousands of parallelized filters onto an equivalent of a single macroscopic filter employed traditionally [286]. The construction of a massively parallel bank of switchable micromechanical filters can then be used to switch between desired frequency bands over a large frequency range as needed, rather than relying on a single tuneable macro filter [286]. Figure 5.1 is a block diagram of a MEMS switchable receiver front end-architecture with each filter corresponding to a unique individual micromechanical filter, the application and removal of a dc-bias voltage provides for a frequency selective device that can potentially enable substantial power savings and reduction in total system area [286]. In order to realise such a system it requires the synthesis of multiple bandpass filter devices with unique frequency transmissions covering the required wide-band range.

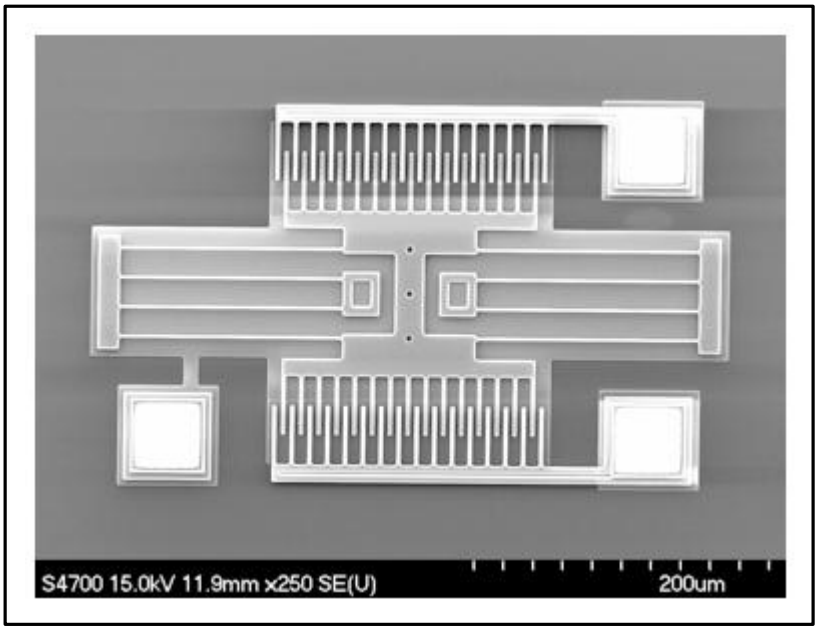


Figure 5.2 SEM of a microelectromechanical folded flexure resonator

Laterally driven folded beam micromechanical resonators such as shown in figure 5.2 offer on-chip alternative devices that provide high-*Q* filtering, reduce size and energy consumption, and are integrated with electrical components directly [131][274]. The device itself consists of a suspended interdigited mass held up by an anchored folded flexure spring component, where the device as a whole is driven and sensed via an electrostatic capacitive comb transducer. An extensive overview on MEMS resonator and oscillator devices is covered in [273].

The coupling of laterally driven microresonators can be used to create bandpass filters, where the resonator is driven by an interdigitated comb transducer while another acts as an electrostatic sensing receiver [273].

A capacitive comb transducer utilises both the mechanical and electrical energies of a system in an equilibrium where an input electrical signal in the form of current or voltage is converted into a mechanical displacement [131][274][287]. These mechanical vibrations are processed through

the mechanical domain of the filter and then converted back into electrical signals by an output sensing comb transducer [287].

The application of a dc-bias voltage V_{pn} and an ac excitation to the one of the resonator-to-electrode comb transducers causes a x directed electrostatic force displacement and with a frequency close to the fundamental resonance frequency of the micro resonator causes the resonator to oscillate. The vibrational energy is transferred from the coupling spring to the output resonator causing it to vibrate, creating a time-varying capacitance between the conductive resonator and the output transducer [274]. Since the dc-bias V_{pn} is effectively applied across the time-varying capacitance at port *n,* a motional output current arises at port *n*, given by

$$i_n = -V_{Pn}\frac{\partial C_n}{\partial t} = -V_{Pn}\frac{\partial C_n}{\partial x}\frac{\partial x}{\partial t} \tag{5.1}$$

Where x is the displacement and $\partial C_n/\partial x$ is the change in capacitance per unit displacement at port *n* [289]. Employing a comb transducer, the displacement of the resonator gives a linear response as the capacitance consists of the overlap capacitance between the interdigited shuttle and electrode fingers [289]. The expression $\partial C_n/\partial x$ can therefore be approximated as

$$\frac{\partial C_n}{\partial x} = \frac{2\xi N_{fin}\varepsilon_o h}{d} \tag{5.2}$$

Where ξ is a constant that models additional capacitance due to fringing electric fields, N_{fin} is the number of finger gaps within the comb drive, ε_o is the permittivity of air, h is the structural layer thickness and d is the comb finger gap spacing [288][131].

It is the output signal which forms the characteristic frequency transmission required for filter signal processing. The central frequency of an individual resonator and filter as a whole is dependent on the resonance frequency of the individuals resonators, the bandwidth characteristics of the filter are dependent upon the coupling springs and its physical properties, in particular its stiffness in the driving direction [273].

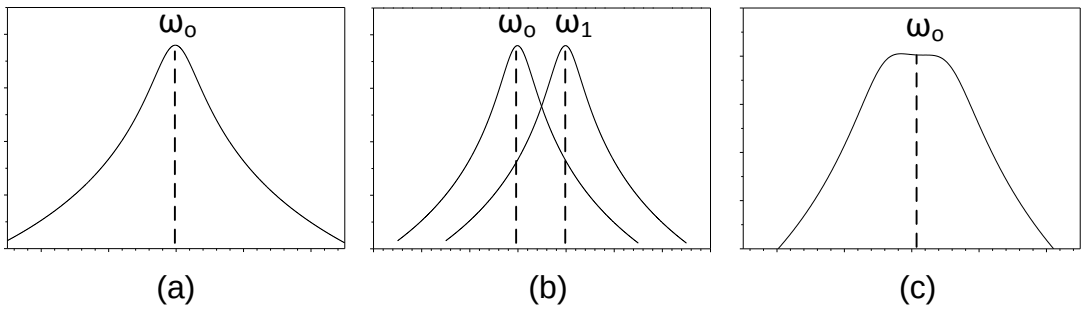


Figure 5.3 Filter frequency characteristics for a (a) single resonator, (b) two separate resonators and (c) a coupled two resonator system

A single folded flexure resonator of the type in figure 5.3 exhibits a biquad frequency response required for a high-*Q* bandpass filter as shown in figure 5.3a. In a network topology of multiple resonators, each resonator exhibits its own unique biquad frequency response as shown in

figure 5.3b and adds it to the coupled system as a whole. The spacing and shape of the filter is affected by a number of variables, in particular coupling spring stiffness, the variation of which can alter the shape to give a desirable flat pass band for a filter of this type as shown in figure 5.3c, additional electronics and techniques such as resistance or *Q*-adjustor components can also be utilized for transmission shape control.

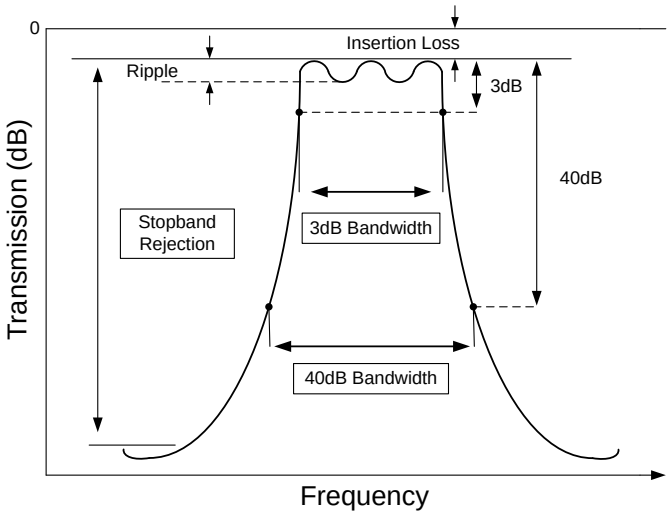


Figure 5.4 Bandpass filter parameter specifications [131]

Some common characteristics of a bandpass filter are shown in figure 5.4 highlighting important components of a successful filter, such as bandwidth, insertion loss, stop band attenuation, and shape factor [131]. This particular filter shape can be readily achieved through the coupling of two or more resonator tanks giving rise to a coupled bandpass biquad network topology. The coupling spring of the mechanical network topology looks to effectively pull the resonator frequencies apart, creating two closely space resonator peaks that constitute the end of the filter pass band [274].

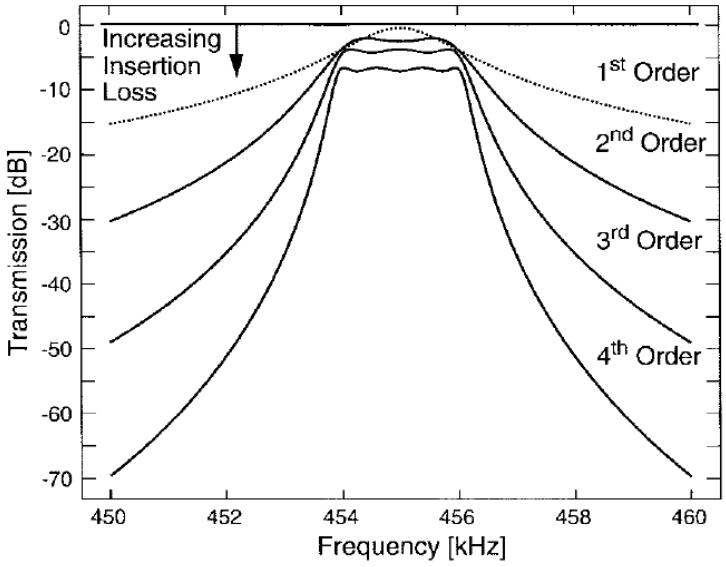


Figure 5.5 Bandpass filter characteristics as a function of increasing filter order taken from [131]

The number of resonators within the coupled filter network constitutes the order of a particular bandpass filter. The order plays an important role in the overall filter characteristics with higher order filters giving a sharper roll off and smaller shape factor leading to higher selectivity, however at a cost of increased insertion loss [131]. An example of a 3rd order filter can be found in [131] and the characteristic change for transmission shape for a number of order systems is shown in figure 5.5.

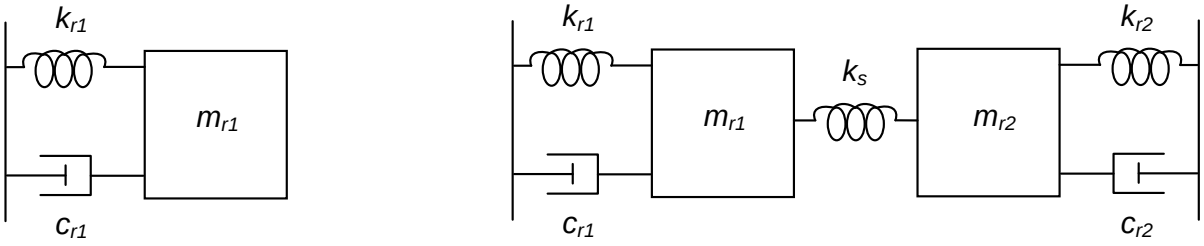


Figure 5.6 Lumped mechanical equivalent of a micro resonator device (a) and a 2nd order bandpass filter (b)

The modelling of a general N-resonator series filter can be accomplished mechanically using a mechanical equivalent circuit of a single resonator device, modelled as a spring – mass – damper system and then through coupling of multiple resonators through the use of a soft mechanical spring [287].

In figure 5.6a a lumped parameter mechanical circuit model of a resonator consists of the components for mechanical mass (m_r), stiffness (k_r) and damping (c_r) and through coupling of a weak spring (k_s) can be configured into a 2nd order bandpass filter as seen in figure 5.2b.

A common approach towards analysis of filter designs is to transform the mechanical elements into their equivalent electrical elements using the analogy modelling method [131][274][287]. Table 5.1 outlines a number of indirect analogies between mechanical and electrical variables.

Table 5.1 Mechanical to electrical equivalence in the force-current analogy

Mechanical Variable	Electrical Variable
Damping, *c*	Resistance, *R*
Stiffness^{-1}, k^{-1}	Capacitance, *C*
Mass, *m*	Inductance, *L*
Force, *f*	Voltage, *V*
Velocity, *v*	Current, *I*
Momentum, *p*	Magnetic flux linkage, ψ
Displacement, *x*	Charge, *q*

The analogous link between the two disciplines means that the force (F) and velocity (*v*) of a mechanical system can be treated as current (*I*) and voltage (*V*) in an electrical system.

The development of equivalent circuit representations is based on the analogy in the mathematical descriptions that exists between electric and mechanical phenomena [289]. Similarities in the equations governing the behaviour of electric and mechanical systems are

where the analogies are drawn and this is illustrated in Newton's second law of motion. Here the relation of force F and velocity u for a rigid mass m, is arranged as:

$$F = m\frac{du}{dt} = m\frac{dx^2}{dt^2} \tag{5.3}$$

and the subsequent electric equivalent an inductor relates as:

$$v = L\frac{di}{dt} = L\frac{dq^2}{dt^2} \tag{5.4}$$

In this analogy the force *F* plays the same role as the voltage *v,* the velocity *u* as the current *i*, and the displacement x as charge q [289].

A series LCR circuit is equivalent to a 1 DOF mechanical mass-spring-damper system. Bandpass filters can be designed through the use of electromechanical analogies, where the electrical domain inductance, capacitance and resistance of a LCR ladder filter can be implemented via analogous values of mass, stiffness, and damping in the mechanical domain [288]. The folded flexure resonators outlined previously can be equated to LCR tanks within the electrical domain, and when coupled together using coupling shunt capacitors form a filter network [288][273].

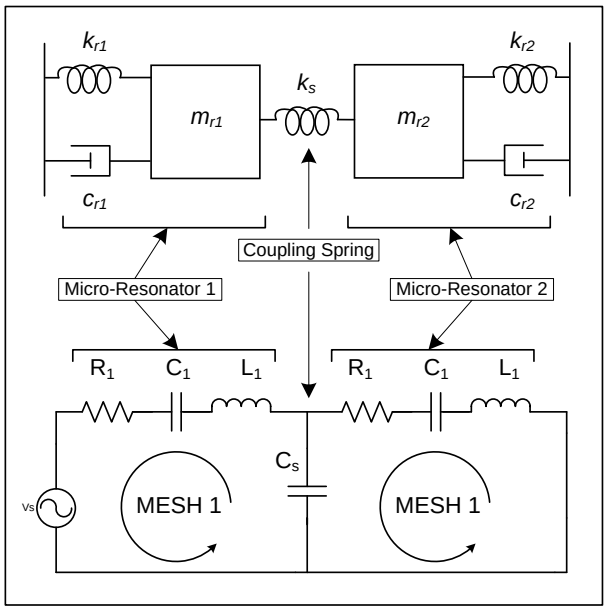


Figure 5.7 (Top) Lumped mechanical equivalent model of a 2nd order bandpass filter with massless coupling springs, (Bottom) corresponding electrical equivalent *LCR* network neglecting static capacitance at the I/O ports and utilizing an *I-type* coupling shunt capacitor.

Looking into an electrical port, with all other ports grounded of the electrical equivalent circuit of figure 5.7, a transformed *LCR* circuit is seen, with element values directly or inversely proportional to the mechanical circuit element values at the shuttle location modified or transformed by the electromechanical coupling parameter η_{en} [274][131]. The approach of relating the electrical equivalent element values with the mechanical equivalent values through transformation, allows the formulation of the electrical equivalent circuit using actual values for mass, stiffness and damping as the values for the inductance, capacitance, and resistance in an *LCR* circuit [131][288].

$$R_{xn} = \frac{c_{rs}}{\eta_{en}^2} = \frac{\sqrt{k_{rs} m_{rs}}}{Q \eta_{en}^2} \tag{5.5}$$

$$L_{xn} = \frac{m_{rs}}{\eta_{en}^2} \tag{5.6}$$

$$C_{xn} = \frac{\eta_{en}^2}{k_{rs}} \tag{5.7}$$

$$\eta_{en} = V_{pn} \frac{\vartheta C_n}{\vartheta x} \tag{5.8}$$

$$\frac{\vartheta C_n}{\vartheta x} = \frac{2 \xi N_{fin} \varepsilon_o h}{d} \tag{5.9}$$

The equations relating the mechanical and electrical values to each other are shown above, equations 5.5 – 5.9. The electromechanical transformation coupling parameter η_{en} allows for the synthesis of an LCR filter to be undertaken entirely within the electrical domain and later converting to equivalent mechanical values for device level synthesis later [274]. Figure 5.8 portrays the equivalencies of the mechanical and electrical modelling domains with a 2D schematic of a 5th order micromechanical filter and its equivalent electrical circuit model and the links between the two.

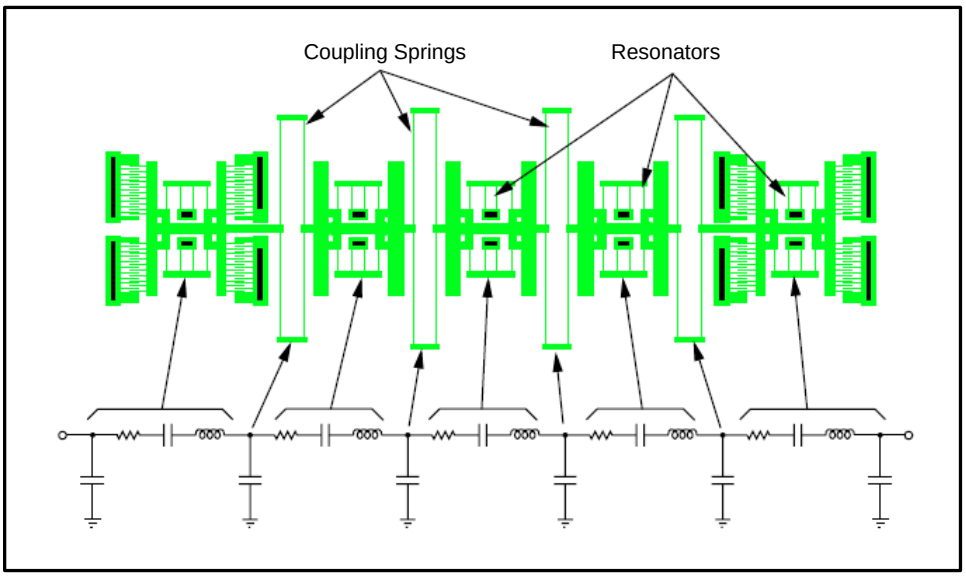


Figure 5.8 2D schematic of a 5th order micromechanical filter, explicitly showing electromechanical equivalencies [288]

The electrical equivalent modelling and mechanical modelling of the bandpass filter device allow for representations to be constructed at the system, device and physical levels, and therefore provides a good case study to investigate each role each level can play in design optimisation of a MEMS device.

5.2 Multi-objective evolutionary algorithm based MEMS bandpass filter synthesis

The application of multi-objective evolutionary algorithms to the design optimisation of MEMS has already been demonstrated both within the literature [43][50][122][124] and in chapter 2

over a number of case studies. In order to investigate the overall performance of multi-level design optimisation strategies towards MEMS a suitable case study that covers the scope of MEMS modelling, analysis and design optimisation often encountered by designers needs to be constructed.

Discussed previously, MEMS bandpass filters, an important component in a number of signal processing systems offers a robust case study into the design optimisation of a MEMS device, with an inherently multidisciplinary nature the device covers all necessary levels of modelling abstraction under investigation in the hierarchy of MEMS design.

MEMS switchable receivers that employ multiple parallelized filter devices require the ability to design individual filters which match certain characteristics, in particular a transmission frequency shape which matches that of a bandpass filter, and a specific designed central frequency for the filter itself.

MEMS bandpass filter design has being undertaken in the past in both a hand driven methodology [131][288] and also through automation [50][53]. These methods employed modelling and analysis paradigms from two levels of MEMS design, the system and the device level. The construction of system level electrical equivalent bandpass filter models allowed for the overall filter characteristics to be analysed using circuit simulators such as Spice [55] or through bond graph representations [59] and ultimately allowing designers to synthesize designs which matched the required filter transmission goals. Using these electrical equivalent solutions it is than possible to extract target objectives for synthesis of realizable 2D layout filter devices, through the use of coupling transformation equations. Therefore the synthesized electrical equivalent circuit models and their constituent values for resistance, capacitance and inductance can be converted to their equivalent mechanical values of damping, stiffness and mass, to be used as objectives for device level layout design.

The design and optimisation of a bandpass filter employs modelling and analysis paradigms from two levels of interest, the system and the device level, and has sufficient scope to include the final physical level of MEMS design as shown in figure 5.9. The application of multi-objective evolutionary algorithms to this particular problem is novel and therefore requires validation in order to establish whether such a stochastic automated approach is viable and robust for the design goals of such a case study.

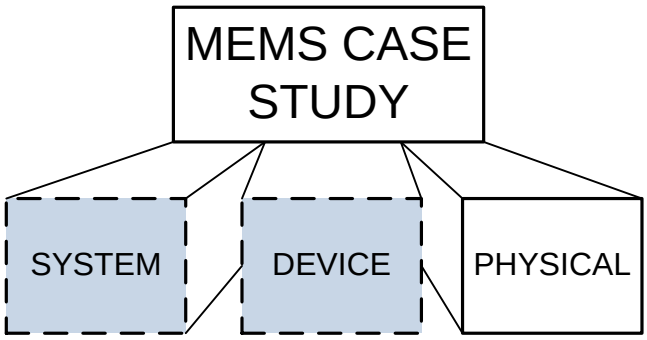


Figure 5.9 MEMS case study hierarchical modelling and analysis levels – two levels utilized in bandpass filter design validation, system and device levels.

The bandpass filter problem can be broken into two individual optimisation problems, a system level bandpass filter design problem, and finally a device level folded flexure design problem, each of which will be expanded upon in the next section.

5.2.1 MEMS bandpass filter synthesis

MEMS bandpass filter synthesis looks towards the creation of a suitable filter transmission shape which meets the specific objectives set out by the designer. Bandpass filters which employ the mechanical vibration properties of a MEMS folded flexure resonator can be reduced to a simple spring-mass-damper model for analysis. Such a model can also be modelled using an equivalent electrical model and the frequency transmission shape can be determined using a suitable circuit simulator such as Spice [55].

The design and optimisation approach utilized looks to couple a multi-objective evolutionary algorithm, in this instance NSGAII [45], with an electric circuit model representation, coined (GAECM). The electric circuit models a bandpass filter outlined previously and consists of a single or series of coupled resonator *LCR* tank components as shown in figure 5.8.

Bandpass filter transmission represents the main qualifier for performance of the designed *LCR* bandpass network; the components of such can be broken down into a set of simple characteristics similar to those in figure 5.4. Characterized in figure 5.10 is an idealized frequency transmission for a band pass filter, where the shape consists of two distinct regions, the pass band and the stop band. In an ideal bandpass filter the pass band, the targeted frequency range where signals are unfiltered, is signified by a completely flat transmission with zero insertion loss and no gain. Outside of the targeted pass band lies the stop band regions, here all incoming signals are attenuated away, in essence filtered, depending on the application this can lay between 20 and 120 dB of the nominal pass band attenuation. Ideally the drop from pass band to stop band is instantaneous.

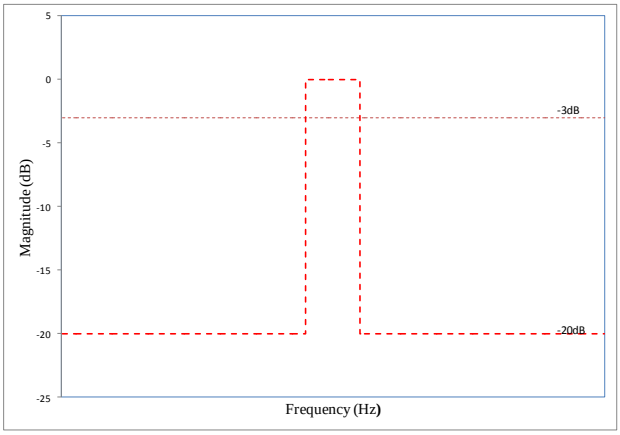


Figure 5.10 Idealized bandpass filter response

In reality no filter is ideal, however this fictitious representation does provide a reasonable template of characteristics in which to evaluate and quantify new filter designs against targets for pass band and stop band frequency ranges.

Therefore two design objectives can be constructed as a means to evaluate new designs created by the optimizer chosen to undertake the design optimisation of the MEMS bandpass filter and are outlined in figure 5.11.

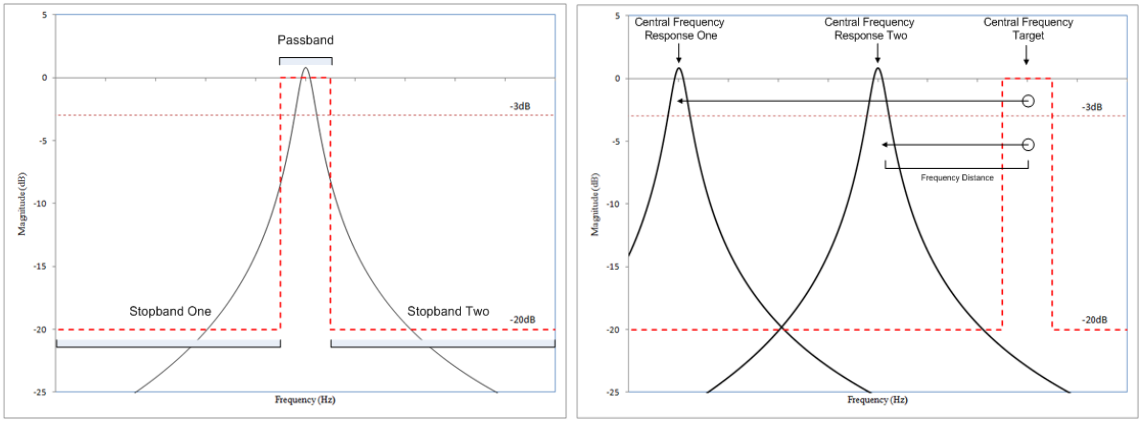


Figure 5.11 MEMS bandpass filter synthesis breakdown for (a) filter objective and (b) central frequency objective

A frequency transmission from a single micromechanical resonator similar to that shown in figure 5.11a consists of a number of frequency data points plotted against the magnitude in units of dB. The quality and performance of the filter transmission can be measured by simply calculating where each data point lies within the pass band and stop band ranges outlined and measured against their target magnitude, in this case $0dB$ for points within the passband and $\leq -20dB$ within the stop band regions. The overall frequency performance can then be quantified as a sum of the total deviation from each of these ranges for the data points within the frequency transmission. Ideally all data points that lie within the pass band will have 0 insertion loss and no gain giving a magnitude of $0dB$, while all points within the stop band will be $\leq -20dB$ and therefore have a deviation of 0 for both regions.

Central frequency of the bandpass filter is important when wanting to design a frequency transmission for a targeted portion of the spectrum. The central frequency of a transmission is simply calculated as the distance of the peak frequency data point to the desired central frequency outlined by the designer. The objective is both a targeted design goal and a guide to the optimizer, allowing individual or coupled resonator transmission responses to move closer to the targeted region of interest.

As outlined in figure 5.7 and schematically in figure 5.8 the components of interest are the individual *LCR* tanks, the coupling capacitor shunts and the values that make up equations 5.8 and 5.9 which are directly related to the comb transducer element of the filter. The individual values and the overall network topology of the circuit model produce the frequency transmission which is then used to evaluate the solution.

In order to effectively simulate and analyse the circuit model solutions produced a circuit simulator is needed for the required AC analysis. Spice [55] is a common analog electronic circuit simulator that allows the constructed electrical equivalent bandpass filter circuit models to be analysed for their frequency transmission, and it has already been utilised in similar filter design synthesis [131][275].

In order to undertake the design optimisation of a MEMS bandpass filter two new modules need to be constructed within the framework outlined in chapter 4, a problem module which holds all problem specific information, and an analysis module which allow for designs within the framework to be simulated and evaluated as shown in figure 5.12.

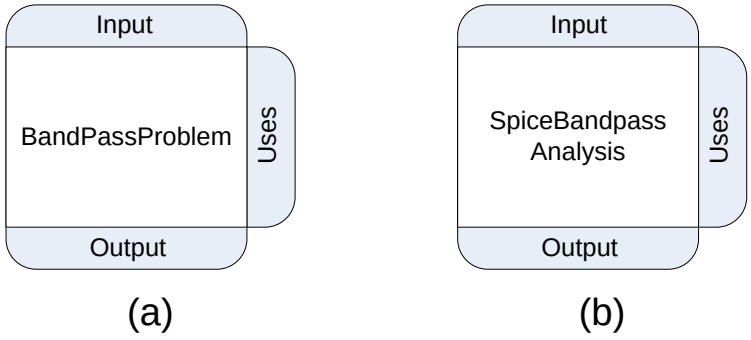


Figure 5.12 Framework modules for (a) bandpass filter problem and (b) bandpass filter analysis

The bandpass filter problem module holds information on the design variables, objectives and constraints needed to successfully synthesis a bandpass filter. A table of the variables and objectives is shown below in table 5.2. The bandpass filter analysis module acts as a link between the design optimisation framework and the simulator software used to analyse the individual solutions. The module constructs a parameter object using the values present in the solution and the uses this to override a bandpass filter netlist file used by the SPICE simulator for analysis. Module specific parameters allow for analysis to occur over any specific range and any number of sampling points and after analysis output data is retrieved for evaluation.

Table 5.2 Bandpass Filter Problem Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Voltage	Real Valued	1	200
Tank Number	Integer	1	9
Finger Number	Integer	1	200
Thickness (μm)	Real Valued	2	30
Capacitance (F)	Real Valued	1E-15	1E-11
Inductance (H)	Real Valued	10	100000
Coupling Spring Capacitance (F)	Real Valued	1E-15	1E-11
Tank	Branch	N/A	N/A
Objectives		Constraints	
Bandpass Filter Response Error	Minimize	N/A	
Bandpass Central Frequency Error	Minimize		

The multi-objective genetic algorithm NSGAII is chosen as the optimizer tasked with evolving and optimising solutions to the bandpass filter problem and an example of the environment pathway with the two new modules integrated into it is shown in figure 5.13.

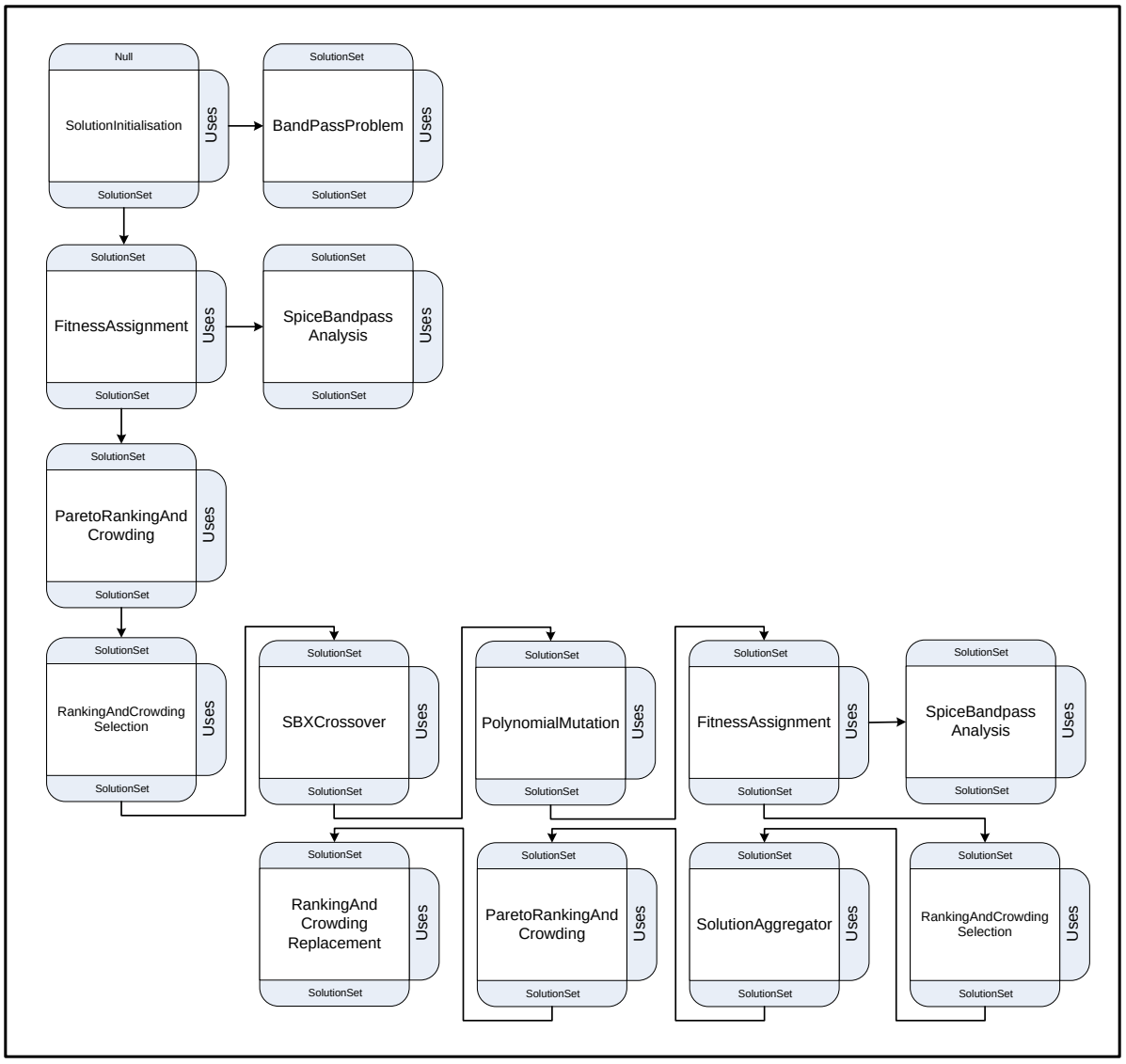


Figure 5.13 MEMS bandpass filter NSGAII environment pathway

The representation consists of real valued and integer variables and is of a variable length, increasing or decreasing in size relative to the tank number of the solution. The representation along with an overview of the design problem is shown in figure 5.14, which also includes reference to the individual node markers and structural tags utilised by the representation and modules within the pathway. A breakdown of the main processes used to create and run the algorithm displayed in figure 5.13 is shown below in algorithm 1.

Algorithm 1: Framework Pseudo Code

1. Execute master control program
    (a) Run queue commands
        i. Request configuration data
        ii. Build framework
            - For each environment data object in configuration data
                - Add environment
                - Set environment tag
                - Load modules
                - Initialise Modules
                - Load environment pathways
                - Initialise environment pathways
        iii. Activate
            - Set main loop as active
        iv. Resume
            - Set master control program state to active
    (b) While main loop is active
        i. While master control program state is active
            - Update queue with eternal commands
            - Invoke queue commands
                - Iterate all environments
            - Increment cycle count
2. Iterate all environments
    (a) For each environment
        i. Execute environment
            - If environment state is active
                - For each pathway in pathway collection
                    - Execute environment pathway
                        - If pathway state is active
                            - For each pathway node in collection
                                - Execute pathway node
3. Execute pathway node
    (a) If pathway node state is active
        i. Chain link start
            - Update Inputs
            - Update Outputs
            - Construct execute pathway
        ii. Chain link execute
            - Execute module function
        iii. Chain link end
            - Retrieve output data object and replace current existing one
            - Clear all output slots of data
            - Clear all input slots of data

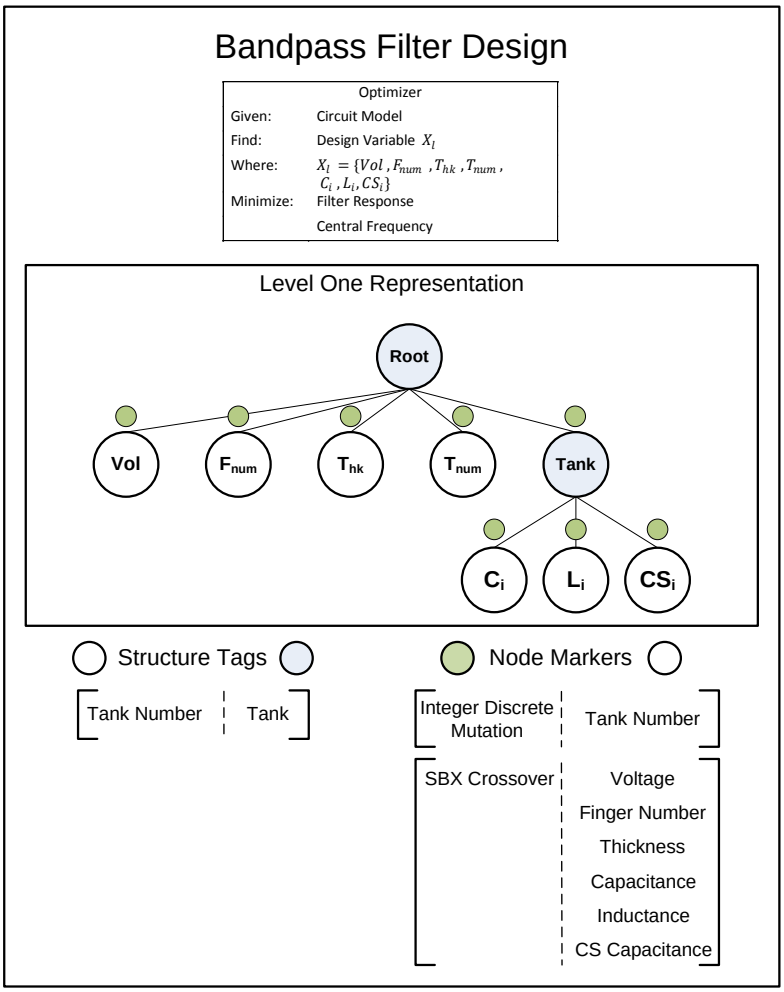


Figure 5.14 Bandpass Filter design template, with overview of problem, default representation, associated structure tags and node markers.

Structural tags relate to specific branch nodes within the representation and the nodes that control the count or number present. In this particular problem representation, the tank number variable and subsequently its value, has control over the tank branch node and increasing or decreasing this value leads to the addition or removal of the specific tank subtree structure. This effect is most evident within the polynomial mutation module within NSGAII, where changing the value of a structure controlling variable such as tank number can lead to effectively cloning or removal of a randomly chosen structure as highlighted in figures 5.15 and 5.16.

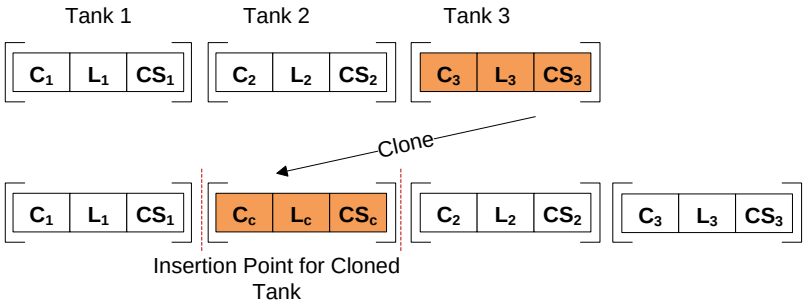


Figure 5.15 Structural cloning of tank component within problem representation

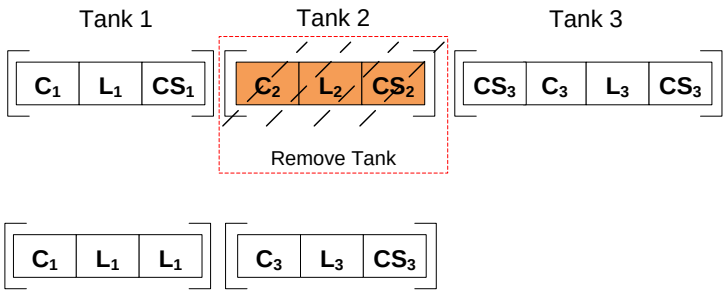


Figure 5.16 Structural removal of tank component within problem representation

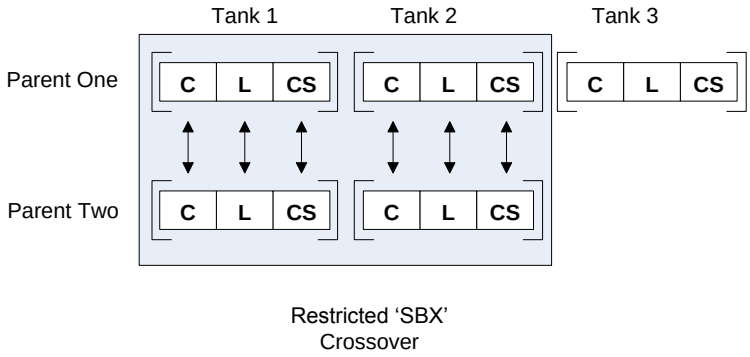


Figure 5.17 Restricted crossover within problem representation

Each node within the representation can contain a set of markers which can be used as some form of identifier or data object. These node markers can act as useful pieces of information or tools which can then be exploited to provide a diverse range of actions depending on the actions of modules that might utilise them. A simple example is shown in figure 5.14 and relates to the module for SBX crossover within the outlined NSGAII environment pathway of figure 5.13. The standard SBX operator within NSGAII has been modified to allow control over what nodes to perform the crossover operation or not, dependent on whether the nodes themselves are marked to do so. Therefore all nodes which have a marker for 'SBX Crossover' have that particular operation performed upon them, while those that don't are ignored. In the bandpass filter design problem the variable for tank number is excluded from SBX crossover, this is as a result of preliminary testing which found the tank number deviated to the mean value early in the design process, and this convergence disrupted the overall exploration for optimal solutions. The final node marker listed is also related to tank number, marked with 'Integer Discrete Mutation', this relates to the polynomial mutation module and how it handles discrete integer values. The polynomial operator of NSGAII has been updated to include discrete values and where appropriate switch to simpler random discrete mutation which lies between the bounds of the chosen variable. In this instance tank number has bounds between 1 and 9, and therefore mutation of the variable is simply a random choice within the bounds of the variable.

A final adaptation within NSGAII and the SBX crossover module is how it handles structured and varied length representations. The representation used for this particular problem is structural in the sense it contains identifiable units which represent the *LCR* tanks used by the electrical equivalent circuit model, and varied, as the number of tanks can change depending on a controlling variable tank number. In order to perform successful crossover over two solutions

with different lengths, as a result of tank number, a restriction is placed on the SBX crossover operation. The solution with the lowest number of structural units, in this case tanks, is chosen as the bounds of crossover, and therefore as shown in figure 5.17 only those tank units within both solutions have the crossover operation performed.

Table 5.3 Case Study Parameter Ranges

	Case Study One	Case Study Two	Case Study Three
Passband	312Hz – 1000Hz	19.5kHz – 20.5kHz	99.5kHz – 100.5kHz
Stopband 1	1Hz – 312Hz	1Hz – 19.5kHz	85kHz – 99.5kHZ
Stopband 2	1000Hz – 10kHz	20.5kHz – 25kHz	100.5kHz – 110kHz
Central Frequency	656Hz	20kHz	100kHz

Table 5.4 Circuit Design Variable Parameters

	Case Study One		Case Study Two		Case Study Three	
Variable Type	Lower Values	Upper Values	Lower Values	Upper Values	Lower Values	Upper Values
Resistance (Ω)	-	-	-	-	-	-
Capacitance (F)	1e-15	1e-11	1e-17	1e-14	1e-18	1e-15
Coupling Spring Capacitance (F)	1e-15	1e-11	1e-17	1e-14	1e-18	1e-15

In order to effectively test the robustness of this design methodology for MEMS bandpass filter synthesis a range of frequencies are chosen. Three bandpass filter case studies are outlined in table 5.3 beginning with a relatively low frequency taken from [53], two more bandpass filter problems are introduced at the 20kHz and 100kHz range.

The design parameters are the same as those listed in table 5.2 except in the case of capacitance and coupling spring capacitance. The central frequency of the bandpass filter is determined primarily by the frequencies of the resonators that make up the network, of which these are heavily influenced by the mass and stiffness of the device [274]. In order to allow synthesis of bandpass filter transmissions which lie within the vicinity of the target central frequency the bounds for capacitance have been varied so initial design lies within the region of interest at the start of the design process as shown in table 5.4. The resistance component of the *LCR* tanks within the electrical equivalent circuit models is calculated using equation 5.5 which is linked to the tanks inductance and capacitance values as seen in equations 5.6 and 5.7 and follows similar approaches outlined in the literature [131]. The quality factor or *Q* value is fixed rather than calculated to a value of 40,000 which is consistent with the fabricated polysilicon folded flexure resonator within a vacuum, and has been outlined in [131].

Each case study as outlined in table 5.3 is fixed to a specific range where points are sampled at specific frequencies and then used to evaluate the two objectives outlined previously as seen in figures 5.11. These were a range of [0Hz-10kHz] for case study 1 resulting in 10,000 sampling points, and [0Hz-25kHz] and [85kHz-110kHz] for case studies 2 and 3 respectively, resulting in 25,000 sampling points. Preliminary testing had discovered that given the size of the stop band in relation to sampling size, frequency transmissions would predominantly converge to having incredibly low magnitudes in such as to avoid having a high error as a result of the stop band

$\leq -20dB$ target limit, as shown in figure 5.11. As a result weighting factors for the sum of the stop bands were set to 'divide' the value by 9 and 25 in order for the algorithm to not focus to heavily on optimising the stop band.

After synthesis of the electrical equivalent bandpass filter circuit models the values can then be converted into their mechanical equivalent values using the equations of 5.5 – 5.9 and then used as objectives for device level design optimisation of a folded flexure resonator. The next section looks at folded flexure design synthesis and how it is to be undertaken within the framework outlined in chapter 4.

5.2.2 MEMS folded flexure resonator synthesis

The components of a bandpass filter include both the micromechanical resonator tank units and the coupling springs used to connect them. The values for an individual micro resonator, mass, stiffness and damping can be extracted from the electrical equivalent circuit model and then used as objectives towards design synthesis of a micromechanical resonator. This allows for the function of the filter to be entirely designed at the system level using circuit models, and then later realized at the device level with an appropriate layout model of the micro resonator device.

A flexible and accurate modelling paradigm is that of NODAL analysis an approach that utilises simple atomic elements such as beams, mass plates or anchors to construct more complex 2.5D devices such as a meandering resonator or ADXL accelerometer as discussed in chapter 2.

It is possible to construct a folded flexure resonator model using such a modelling approach, in particular the analysis tool Sugar [66] already implemented successfully previously. Figure 5.18 shows an example of a folded flexure resonator built using the Sugar tool, broken into the central mass and folded flexure components and based upon poly-silicon material.

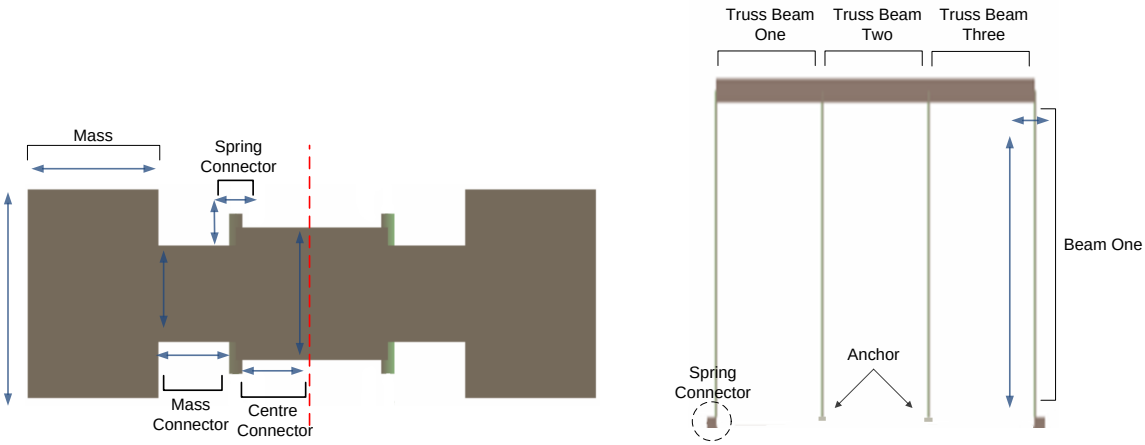


Figure 5.18 SUGAR folded flexure resonator broken down into (a) central mass unit and (b) folded flexure unit

Once again in order to perform folded flexure resonator synthesis within the framework, problem and analysis modules need to be created. Figure 5.19 shows each of the additional modules created along with a new structural crossover module, an operator designed specifically for structured representations like those employed in previous MEMS design

[43][124]. Adaptations to previous crossover operators for the design optimisation of MEMS devices which utilised suspended spring structures like the folded flexure looked to crossover whole components rather than individual variable values as is present in traditional SBX crossover. This involved either the transfer of whole suspended spring structures or their constituent beam parts. In this particular representation only whole spring crossover is undertaken as each spring constitutes only one beam element.

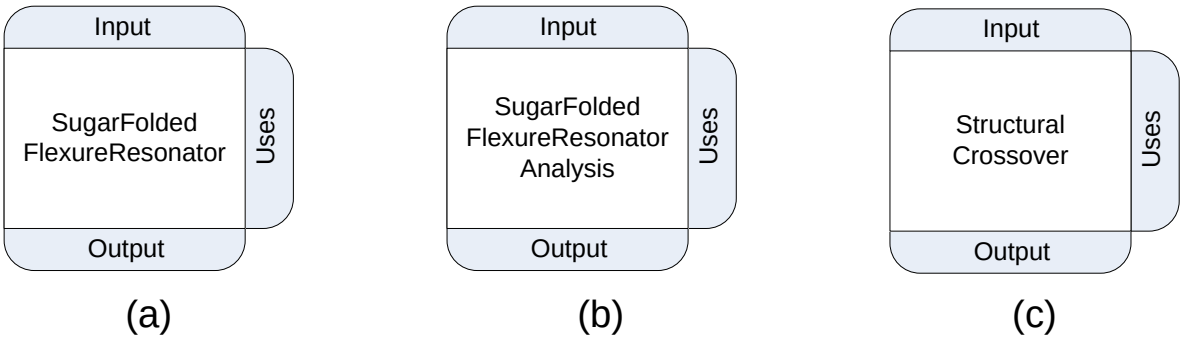


Figure 5.19 Framework modules for (a) sugar folded flexure resonator problem and (b) sugar folded flexure resonator analysis (c) structural crossover

Table 5.5 Folded Flexure Resonator Problem Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Mass Length (μm)	Real Valued	10	600
Mass Width (μm)	Real Valued	10	600
Mass Connector Length (μm)	Real Valued	20	400
Mass Connector Width (μm)	Real Valued	20	400
Centre Connector Length (μm)	Real Valued	55	400
Centre Connector Width (μm)	Real Valued	20	400
Spring Connector Length (μm)	Real Valued	20	50
Spring Connector Width (μm)	Real Valued	20	50
Beam Length (μm)	Real Valued	10	700
Beam Width (μm)	Real Valued	2	20
Anchor Placement Length (μm)	Real Valued	0	400
Objectives		Constraints	
Folded Flexure Stiffness Kx Error	Minimize	N/A	
Folded Flexure Mass Error	Minimize		

The folded flexure resonator design variables and objectives are shown in table 5.5 and consist of real-valued variables for the central mass and folded flexure components of the resonator. The objectives focus upon the mechanical relations of the resonator to the *LCR* tanks in the electrical equivalent circuit model and therefore look to evolve designs which match target mass and stiffness values. Damping and its electrical equivalent *resistance* is not a design target for this particular synthesis problem. The relationship between damping and the mass and stiffness has been outlined in equations 5.5 – 5.9 and for simplicity is treated as relational with regards to mass and stiffness rather than as a separate entity which can be designed to. Damping, essentially removal of energy from the system, can come from many sources, though for this resonator in particular the dominating damping force comes from gas damping. However when packaged within a vacuum this becomes reduced and other damping factors begin to play a

larger role. Though it is possible to synthesis a resonator in relation to damping when employed in atmospheric conditions as was demonstrated in chapter 2, calculating other factors such as thermal stress, parasitic capacitance or anchor loss [290] is more problematic.

Analysis of individual solutions is undertaken in the folded flexure resonator analysis module which acts as an interface with the NODAL simulator Sugar [66]. In the same approach as outlined in chapter 2, a netlist is constructed which employs a template model of a folded flexure resonator which can then be passed a parameters file which contains values that override the default ones to create a new design. A number of analysis calls can then be initiated and the output results retrieved for use in objective and constraint calculation later.

Once again the multi-objective genetic algorithm NSGAII is chosen for the design synthesis of a folded flexure resonator and an example of the environment pathway with the three new modules integrated into it is shown in figure 5.20.

The representation consists of real-valued variables and is off a fixed length and is shown in figure 5.21 along with additional node markers. The design synthesis uses structural crossover for the folded flexure spring components and the standard SBX crossover operator for all other variables and therefore only those nodes marked for SBX crossover have the operation invoked. SBX crossover of the central mass between two individuals is highlighted in figure 5.22.

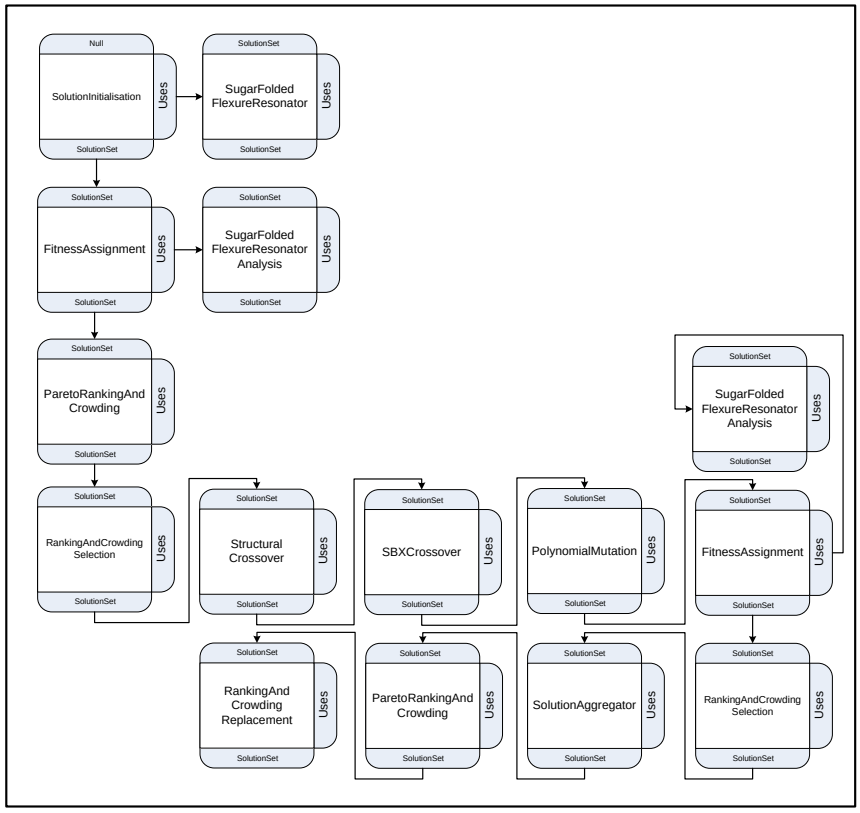


Figure 5.20 MEMS folded flexure resonator NSGAII environment pathway

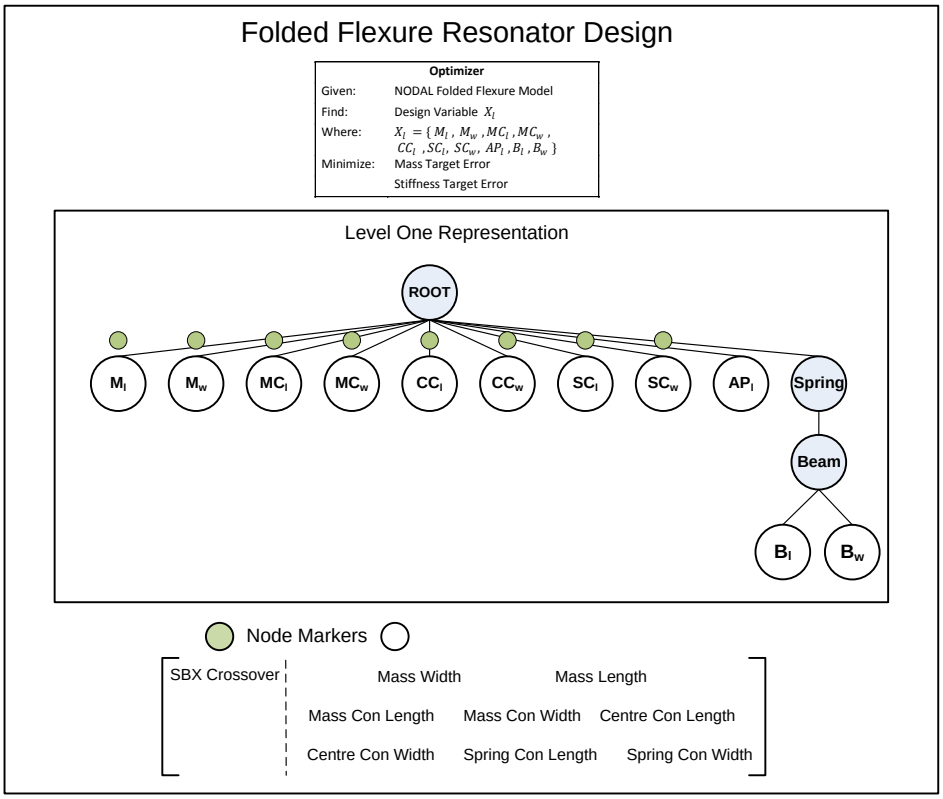


Figure 5.21 Folded flexure resonator design template, with overview of problem, default representation, associated structure tags and node markers.

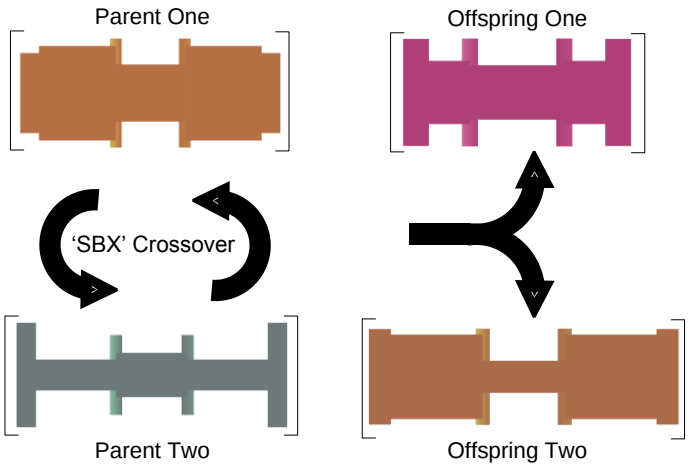


Figure 5.22 Central mass SBX crossover

In both sets of experiments as outlined previously, the multi-objective genetic algorithm NSGAII is used to evolve designs for system level MEMS bandpass filter and device level folded flexure resonator synthesis and optimisation. The parameters for the components of the NSGAII algorithm are shown in table 5.6, in this instance the system level mutation is set to a higher probability of invocation in order to facilitate more structural search as a result of adding or removing '*LCR*' tanks. Two different population sets were explored for system level design, a standard 100 / 100 population and offspring ratio and a smaller 20 / 10 ratio, with both sets undertaking 10,000 functional evaluations. Each case study for both population sets consisted of

five differently seeded tests, while device level design synthesis was tied to the final filter results for each case study and therefore is linked to the number of resonator tanks within each electrical equivalent circuit model.

Table 5.6 NSGAII Parameters

NSGAII	System	Device
Probability of SBX Crossover	0.8	0.8
Probability of Structural Crossover	N/A	0.8
Probability of Mutation	0.35	0.10
Distribution Index for crossover	20	20
Distribution Index for mutation	20	20
Population Size	100 / 20	100
Offspring Size	100 / 10	100
Selection Size	100 / 10	100
Generations	100 / 1000	100
Tests	5	-

5.2.3 MEMS bandpass filter synthesis numerical results, analysis and validation

The case study results for each bandpass filter are presented below, consisting of the best filter transmission results for each population set, along with a table of their objective values. A full set of filter transmission results for each test are held in appendix B. The combined population set results for each test are also shown and the best filter result from each case study is converted into its mechanical equivalent for each resonator and used as objective targets for folded flexure design synthesis with the evolved filter designs also shown.

Table 5.7 Bandpass Filter Synthesis Case Study 1 Results

Best Result Case Study 1: Population 100				
Test	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	941.76	110	112.5	2
2	953.40	86	161.7	2
3	565.25	293	66.4	3
4	478.65	24	43.9	3
5	942.03	256	159.7	2
Best Result Case Study 1: Population 20				
Test	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	940.47	112	1	2
2	1974.60	97	32.70	2
3	476.76	240	7.28	3
4	2130.29	0	109.85	2
5	2130.30	1	108.75	2
Individual Folded Flexure Resonator Values		Best Result Case Study 1		
Tank 1	Equivalent Mass (kg)	5.92e-9		
	Equivalent Stiffness (N/m)	0.083		
Tank 2	Equivalent Mass (kg)	4.78e-8		
	Equivalent Stiffness (N/m)	0.073		
Tank 3	Equivalent Mass (kg)	3.03e-8		
	Equivalent Stiffness (N/m)	0.281		

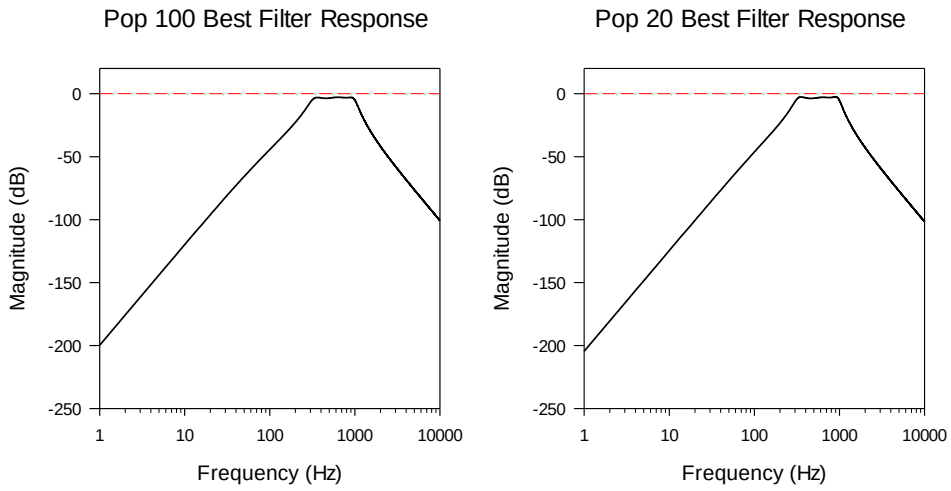


Figure 5.23 Bandpass filter validation 656 Hz best filter response ranked by filter frequency objective for NSGAII (left) 100 population and (right) 20 population sets

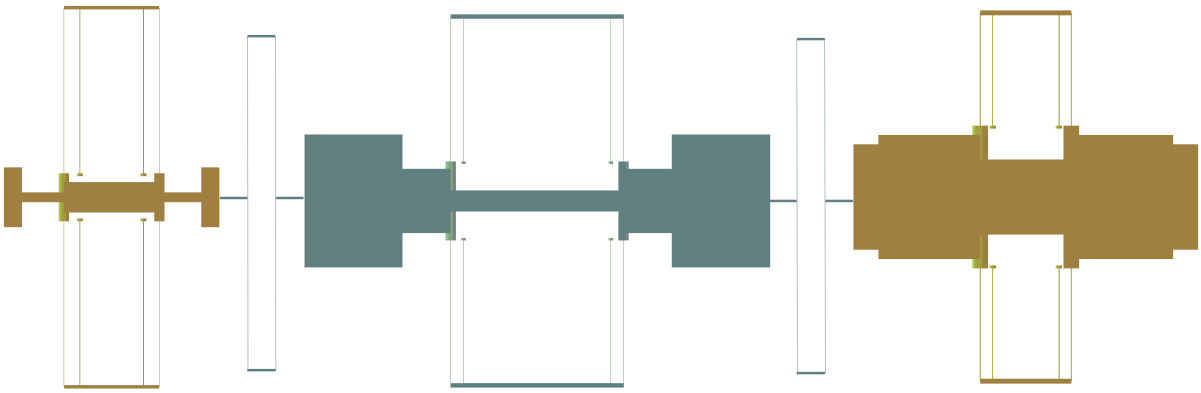


Figure 5.24 Folded flexure resonator layout designs for best result case study one filter

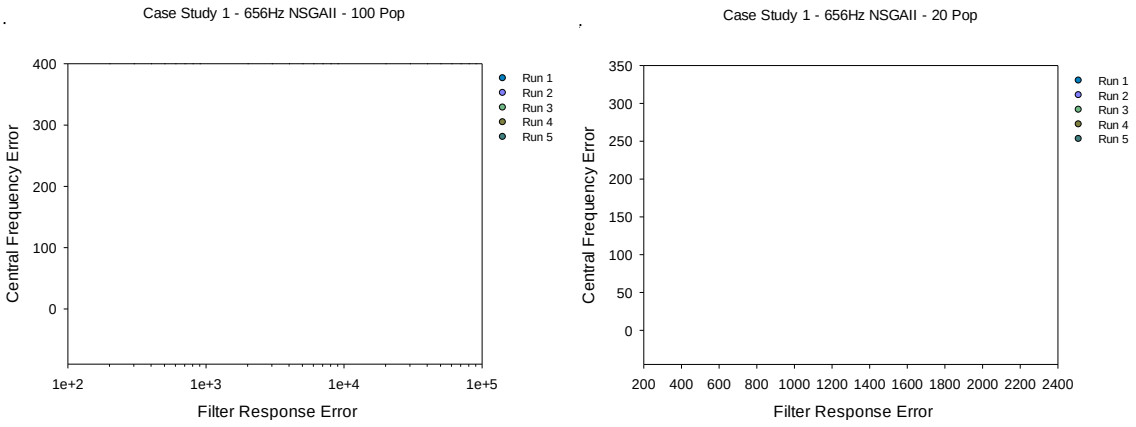


Figure 5.25 Bandpass filter validation 656 Hz run 1 – 5 final population sets for NSGAII (left) 100 population and (right) 20 population sets

The task for synthesizing a 656 Hz bandpass filter in case study one proved the most successful of all the case studies. Both population sets provided very good filter transmissions, however going over the filter transmission results for all tests and table 5.7 it is clear that the higher population set was more robust, providing similar and superior bandpass filter transmissions over the 5 tests. The mechanical equivalent of the circuit model filter values for test 4 of the 100 population set for case study one is shown in table 5.7 and the evolved design in figure 5.24. All three resonator designs had an error from the target objectives of ≤ 0.1% and an example of the average population objective values over the generations for one of these resonators is shown in figure 5.26.

Both sets of experiments yielded populations which have clearly converged to a front, be it around 1 kHz frequency error or higher at 2 kHz. The best designs for filter frequency have come from population sets which contain higher central frequency error and have less convergence, and also exhibit some form of topological search with higher tank numbers.

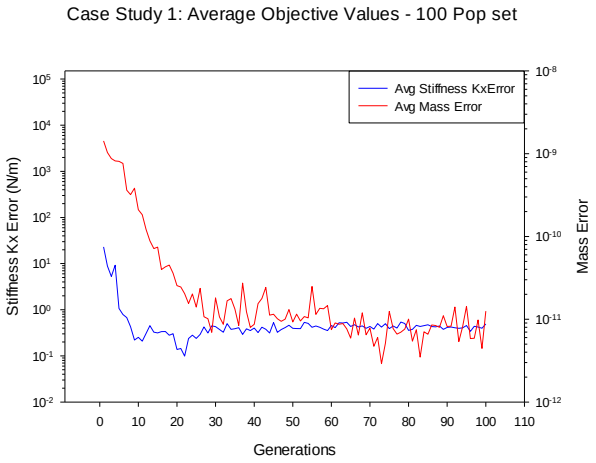


Figure 5.26 Average objective values for population over generations for the design synthesis of a folded flexure resonator

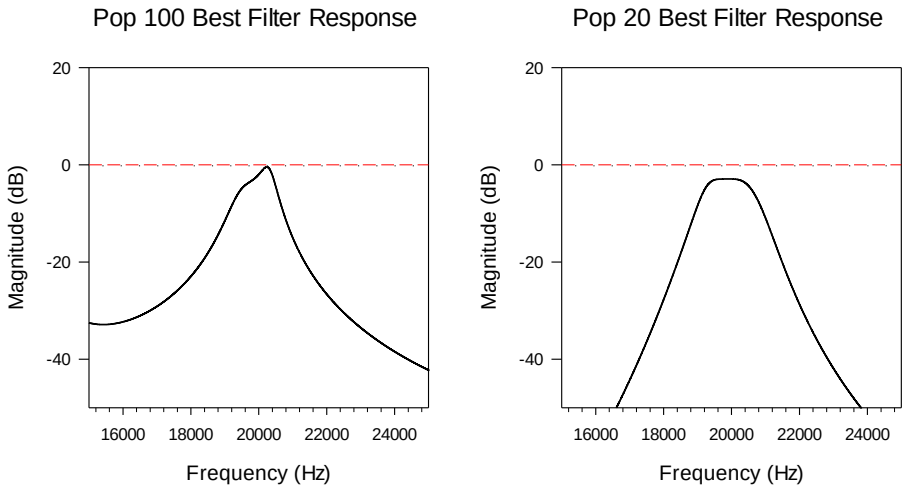


Figure 5.27 Bandpass filter validation 20 kHz best filter response ranked by filter frequency objective for NSGAII (left) 100 population and (right) 20 population sets

Table 5.8 Bandpass Filter Synthesis Case Study 2 Results

Best Result Case Study 2: Population 100				
Test	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	1798.99	230	84.3	3
2	2259.23	1250	54.99	5
3	1990.79	30	16.98	3
4	3085.71	50	102.68	2
5	2422.73	190	2.43	3
Best Result Case Study 2: Population 20				
Test	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	988.58	100	44.16	5
2	1293.24	260	78.03	5
3	2998.03	10	45.62	2
4	2095.91	150	115.56	3
5	1048.50	210	26.65	3
Individual Folded Flexure Resonator Values		Best Result Case Study 2		
Tank 1	Equivalent Mass (kg)	2.34e-10		
	Equivalent Stiffness (N/m)	3.91		
Tank 2	Equivalent Mass (kg)	2.50e-10		
	Equivalent Stiffness (N/m)	3.24		
Tank 3	Equivalent Mass (kg)	2.67e-10		
	Equivalent Stiffness (N/m)	3.99		
Tank 4	Equivalent Mass (kg)	2.77e-10		
	Equivalent Stiffness (N/m)	3.99		
Tank 5	Equivalent Mass (kg)	2.26e-10		
	Equivalent Stiffness (N/m)	3.92		

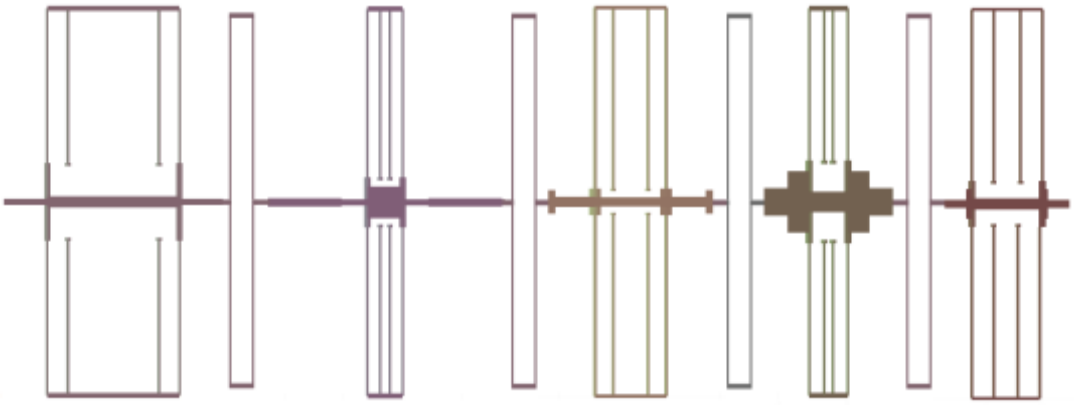


Figure 5.28 Folded flexure resonator layout designs for best result case study two filter

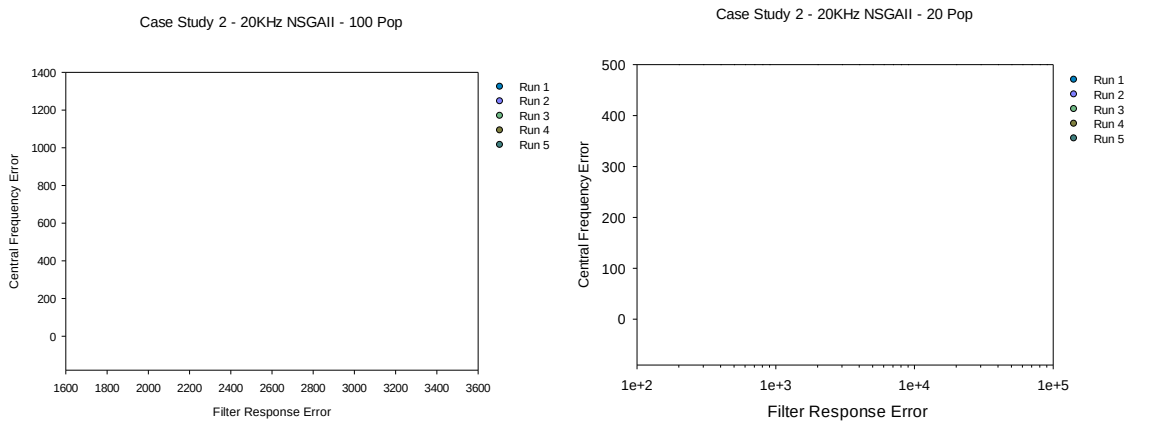


Figure 5.29 Bandpass filter validation 20 kHz run 1 – 5 final population sets for NSGAII (left) 100 population and (right) 20 population sets

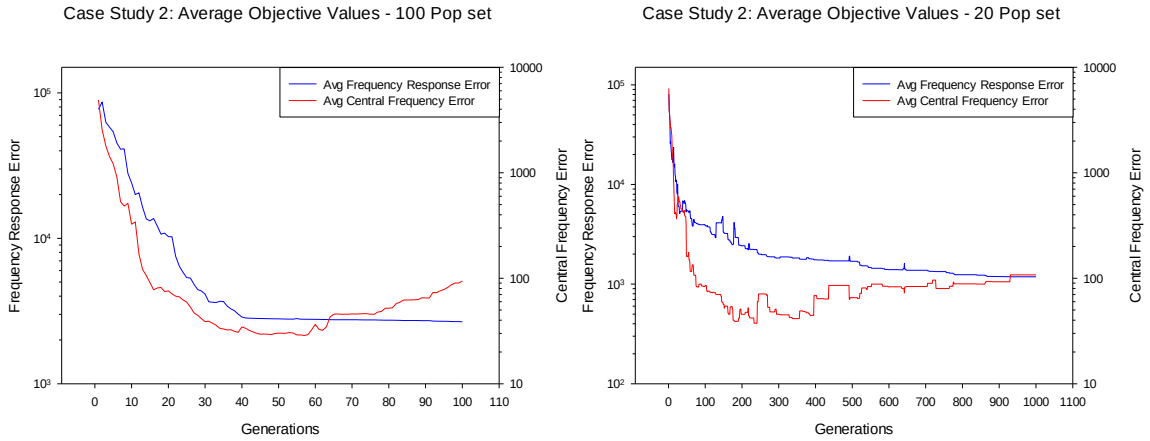


Figure 5.30 Case study two average objective values for population over generations for test 4 population 100 (Left) and test 5 population 20 (Right) sets.

The second case study lies within the 20 kHz frequency region and also includes a much larger pass band then the previous example. This shift has proved much harder as seen in figure 5..27 and table 5.8, resulting in both population sets from providing reasonable bandpass transmissions over each of the 5 tests comparatively to the last case study. The population 20 set proved the most successful with 3 reasonable bandpass transmissions, in particular the best example for filter frequency in test 1. In contrast the 100 population set only provided 2 frequency transmissions with characteristics of a bandpass filter, but with distorted pass band shapes.

The mechanical equivalent values for the best result ranked by filter frequency are shown in table 5.8, here five tanks make up the mechanical filter, each containing similar mass and stiffness values. The evolved resonators are shown in figure 5.28 and once again have mechanical values with an error of ≤ 0.1% from the target objectives set out.

The final population sets for each test are shown in figure 5.29, with the population 100 sets showing less convergence then those of the 20 population. Looking over both population sets those within the 100 population show a broader range of filter frequency error while those within the 20 population sets are more constrained with a much smaller derivation between

individuals within the same population set and a more stable Pareto front. An example of the average objective values for the population over generations is shown in figure 5.30 for both 100 and 20 populations. Both exhibit similar characteristics with a fast linear reduction in error for both objectives for about a quarter of the generations until the average error tails off and either flattens or shows a marked slowing down in error reduction, or in the case of the central frequency begins to rise near the end.

Table 5.9 Bandpass Filter Synthesis Case Study 3 Results

Best Result Case Study 3: Population 100				
Test	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	1632.81	170	86.87	6
2	2405.76	40	31.78	2
3	2712.51	110	169.61	2
4	2087.13	40	152.39	2
5	2289.03	30	197.81	5
Best Result Case Study 3: Population 20				
Test	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	2319.79	40	127.72	2
2	2181.26	30	40.30	2
3	1672.20	10	66.03	3
4	1628.61	20	27.54	3
5	1304.11	190	22.17	9
Individual Folded Flexure Resonator Values		Best Result Case Study 3		
Tank 1	Equivalent Mass (kg)	3.92e-10		
	Equivalent Stiffness (N/m)	160.52		
Tank 2	Equivalent Mass (kg)	4.15e-10		
	Equivalent Stiffness (N/m)	159.72		
Tank 3	Equivalent Mass (kg)	4.03e-10		
	Equivalent Stiffness (N/m)	159.74		
Tank 4	Equivalent Mass (kg)	3.92e-10		
	Equivalent Stiffness (N/m)	160.52		
Tank 5	Equivalent Mass (kg)	2.95e-10		
	Equivalent Stiffness (N/m)	159.74		
Tank 6	Equivalent Mass (kg)	3.90e-10		
	Equivalent Stiffness (N/m)	159.74		
Tank 7	Equivalent Mass (kg)	4.18e-10		
	Equivalent Stiffness (N/m)	158.80		
Tank 8	Equivalent Mass (kg)	4.11e-10		
	Equivalent Stiffness (N/m)	160.52		
Tank 9	Equivalent Mass (kg)	4.07e-10		
	Equivalent Stiffness (N/m)	159.74		

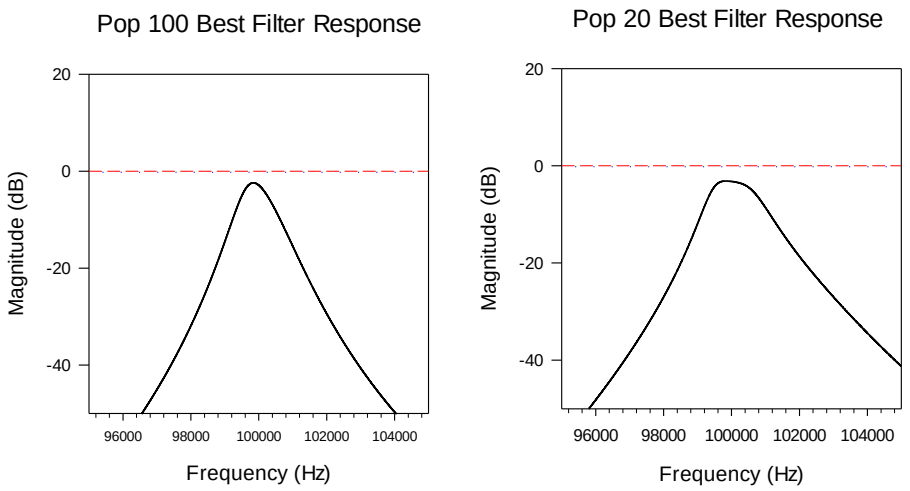


Figure 5.31 Bandpass filter validation 100 kHz best filter response ranked by filter frequency objective for NSGAII (left) 100 population and (right) 20 population sets

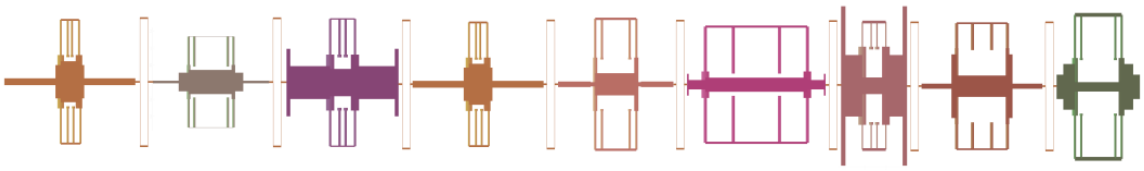


Figure 5.32 Folded flexure resonator layout designs for best result case study three filter

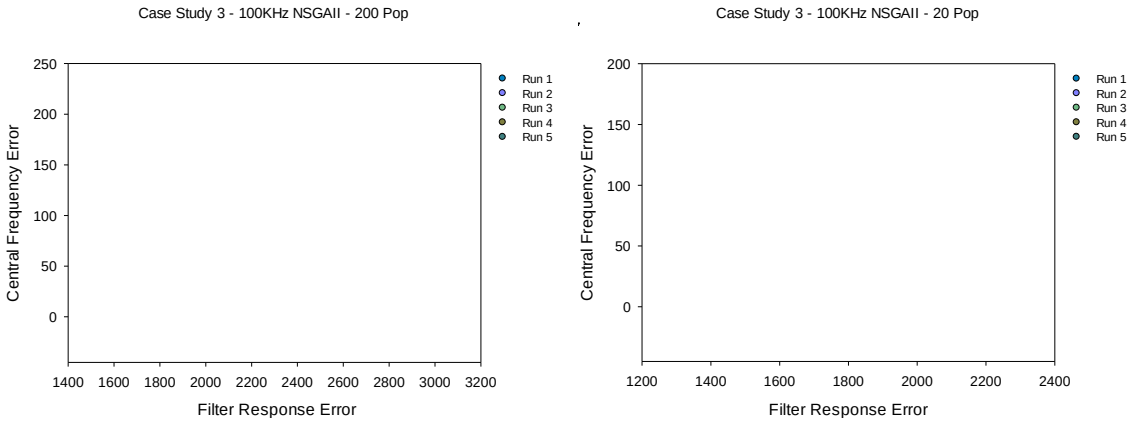


Figure 5.33 Bandpass filter validation 100 kHz run 1 – 5 final population sets for NSGAII (left) 100 population and (right) 20 population sets

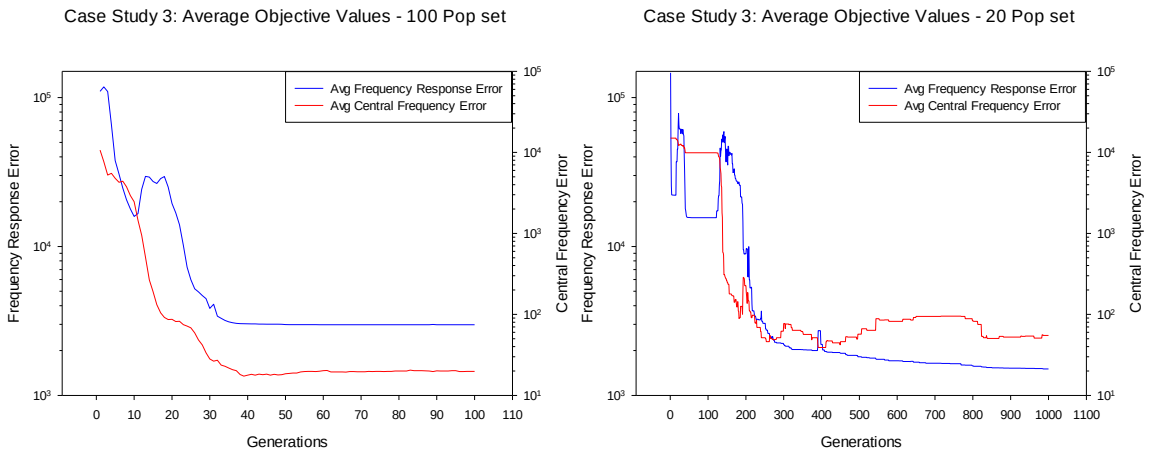


Figure 5.34 Case study three average objective values for population over generations for test 4 population 100 (Left) and test 5 population 20 (Right) sets.

The final case study results provide a similar response to that of the previous case study, both population sets were able to provide bandpass filter transmissions, however the population 20 set was the most successful. The best design ranked by the filter frequency objective was evolved using the mechanical equivalent values calculated and shown in table 5.9 and the resonator layouts are shown in figure 5.32, once again all resonators an error of ≤ 0.1% of the target objectives.

The final population sets show similar convergence onto a front for tests of the population 20 set while the 100 population set is more varied and dispersed for tests 1 and 5, while tests 2-3 show convergence to a front. The average objective values for population over generations for this case study is shown in figure 5.34 for the best tests of both population sets. Similar to the previous case study there is s sudden drop in error on both objectives followed with a slow plateau in the second half of the design process.

5.3 Discussion

The design synthesis of electrical equivalent circuit models for a number bandpass filters using a new multi-objective evolutionary algorithm approach (GAECM) has been demonstrated and validated within the framework outlined in chapter 4. Additional design synthesis of the micro mechanical folded flexure resonators that make up the components of the filters evolved by the GAECM approach has also been undertaken and successfully validated.

The goal of developing a multidisciplinary case study that covered the three main levels of the MEMS design process and provides reasonable difficulty in its optimisation is fulfilled with the creation of the bandpass filter design problem. Currently including both system and device level modelling, analysis and design, the problem has scope to allow expansion into the final level of interest in physical design.

Evaluating the GAECM approach over the three case studies shows an ability to find frequency transmissions which match the targeted bandpass characteristics outlined, however with depreciation in quality as the frequency range is increased over the case studies. One explanation for this can be assigned to the initial start of the design process, and where the

optimisation starts within the design search space. The chosen upper and lower bounds for the inductance and capacitance plays an important role in where the central frequency response of the filter lies within the frequency range.

This is highlighted in figure 5.35, where for each case study, 10,000 randomly initialised solutions are created and analysed, and a histogram of their central frequency is plotted showing the individual count and where they lie for a given frequency. Beside these plots the best solution ranked by filter frequency from the 10,000 random solutions is shown.

The 656 Hz bandpass filter of case study 1 highlights how successfully chosen upper and lower bounds can give rise to randomly initialised solutions which lie within the passband region of interest and provide solutions with good bandpass frequency transmissions. As the range of interest changes to 20 kHz and then 100 kHz it can be seen that for the bounds chosen for the design variables the solutions produced lie further away from the target passband ranges. The quality of the solutions is also lower with limited bandpass transmission shape, with case study two lying close to the target central frequency but consisting of a single resonator, while for case study three the best random solution lies even further away from the target central frequency.

Even so, with the drawbacks related to starting bounds on specific design variables like tank capacitance, the GAECM approach was able to provide comparable bandpass filter transmissions to those within the literature for the current state of the art in automated design [53]. This comes at a significant reduction in functional evaluations, 10,000 compared with 2.6 million, however it is hard to make a direct comparison as both approaches have their benefits and drawbacks, in particular the GPBG method outlined in [53] utilizes bond graphs which are open to more creative design then a rigid *LCR* circuit model.

The electrical equivalent circuit model itself is also a simplified representation, with for example drawbacks in modelling as a one-port imprudence device rather than two-port [274], and representing the coupling shunt capacitors as massless ideal springs when in reality there is some finite mass which may affect filter transmission [275].

There are also additional components which can be added to the filter device which can aid in the overall bandpass transmission post fabrication. Two of these are the addition of *Q*-controllers and frequency tuning. *Q*-controlling resistors placed at input and output ports can be used to flatten the pass band of the filter transmission to more ideal bandpass characteristics, while it is possible to place additional comb drives which are orthogonal to each resonator and through application of a DC tuning voltage alter the electrostatic spring constants and in effect shift central frequency of the filter transmission. Due to fabrication errors and variation the designed filter response of the synthesised device may not match up to the original *in-silica* modelling and analysis. The ability to tune any shifted frequency transmissions post fabrication to the original desired shape is an ideal function [275].

Overall however the use of electrical equivalent to mechanical equivalent conversion methods has proved successful and it provides a direct link between the system and device level optimisation through physical mechanical targets for device layout optimisation leading to a realization of the functional circuit models. This realization was demonstrated with the synthesis of a number of filters and their individual folded flexure resonators using an adapted multi-

objective evolutionary algorithm NSGAII. In all of the folded flexure resonator design synthesis undertaken, there were evolved solutions which yielded 2D layout designs with the objective values for mass and stiffness having an error ≤ 0.1% of the targets specified.

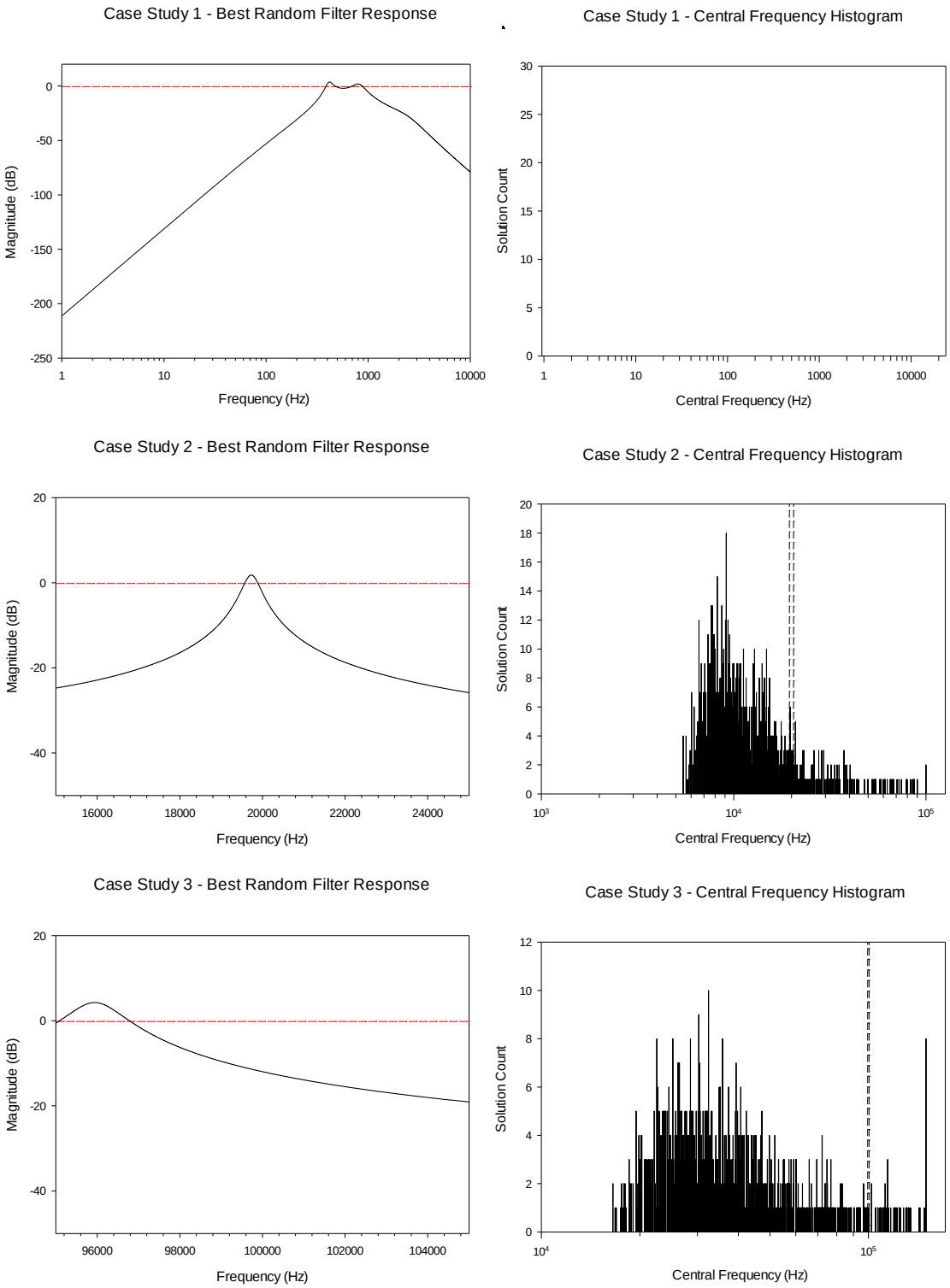


Figure 5.35 – Best random filter response and central frequency histogram for band pass filter case studies, dashed lines mark pass band regions for each filter

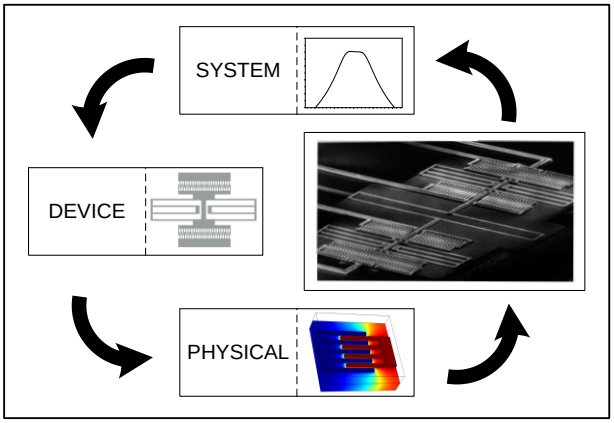


Figure 5.36 – Microelectromechanical bandpass filter case study

This synthesis involved the integration of the nodal simulator Sugar [66] into the design framework and coupling it with the NSGAII environment pathway. Additional variation operators were added to aid search, in this approach through structural crossover, and only requiring 10,000 functional evaluations it compares well with the current state of the art [53] which required 137,500 functional evaluations and was not truly multi-objective.

The micromechanical bandpass filter device can play an important role in many signal processing systems. It is highly multidisciplinary, encompassing system level electrical disciplines and device and physical level mechanical disciplines. The scope and complexity of the device provides a suitable case study for design synthesis and optimisation, and provides the necessary levels of modelling and analysis to evaluate the role multi-level design optimisation can play in the design of MEMS as seen in figure 5.36.

5.4 Summary

Introduced in this chapter is a novel approach for the synthesis of MEMS bandpass filters. Incorporating an electrical equivalent modelling and analysis approach with a multi-objective evolutionary algorithm routine provides for the optimal design of a number of MEMS bandpass filters at a significant reduction in computational cost over the current state of the art. Work now shifts to evaluation of multi-level design optimisation strategies and their application to MEMS design synthesis and optimisation. Each of the three levels (System, Device and Physical) is included, either as individual or coupled design optimisation problems.

6

Uni-Level Design Optimisation

This chapter focuses upon the design synthesis and optimisation of a number of MEMS design problems covering several levels of the hierarchical design process. Three modelling and analysis levels are investigated, with a system level bandpass filter, device level folded flexure resonator and finally a physical level coupling spring. Both the current state of the art algorithms for automated design of MEMS, single level MOEAs, and new set of multi-level and multidisciplinary optimisation algorithms are applied and compared.

6.1 Experimental parameter selection

Experimentation often gives rise to further insight into the design problem that is under investigation. Further analysis of the results can point to mistakes in the original planned setup, or highlight advantageous corrections which might yield better results.

Information learnt while undertaking a single level optimisation strategy might than be utilised later on in other multi-level design optimisation strategies. Though this can be a natural approach to research and design as a whole, in order to not bias one particular strategy over another it is something to be avoided or watched with a cautious eye. Therefore strict guidelines on how each strategy is approached when undertaking the global design problem need to be adhered to.

Two of the most commonly used multi-objective genetic algorithms (NSGAII [45] and SPEA2 [132]) within the literature have been established and validated within the constructed framework of chapter 4. Both of these algorithms are used in the single and multi-level strategies outlined in chapter 3, and each are heavily influenced by the parameters that control them and their constituent operators. In order to affect a fair investigation over the experiments and design strategies to be outlined, the design parameters should remain fixed and consistent where possible.

Decision on the values for the various algorithmic parameters can come from the literature or through experimentation. The algorithmic parameters of interest are:

- Population Size
- Offspring Size
- Selection Size

- Replacement Size
- SBX Crossover Probability
- Polynomial Probability
- Distribution Index Values

The population and offspring sizes used for the default setup are shown in table 6.1 and are consistent with previous experimentation in chapters 2, 4 and 5. The sizes are large enough to stop any premature convergence of the genotype though not excessively so, and therefore do not require a large number of functional evaluations for one generation. Both selection and replacement sizes once again mirror previous experimentation and do not place a heavy selective pressure from generation to generation.

Table 6.1 Default Algorithm Parameters

Algorithm Parameter	Default Value
Population Size	100
Offspring Size	100
Selection Size	100
Replacement Size	200
SBX Distribution Index	20
Polynomial Mutation Distribution Index	20

Adaptation and variation are key components of Darwinian evolution, and one of the main tenets of evolutionary algorithms. Variation within the evolutionary algorithms chosen comes from the operators for crossover and mutation, which are themselves influenced by their individual values for probability of invocation. Within the literature there are reasonable ranges for both, often small for mutation at around 1% to 5%, while crossover is much larger, at around 70% to 90%. A small parametric investigation into the effect the probability of invocation has on design optimisation of MEMS was undertaken in order to define the optimal default parameters for each operator. It is not possible to generalise over the various design optimisation problems faced in MEMS synthesis. However of the bandpass filter under experimentation, the main component, the folded flexure resonator is a reasonable choice of model to explore variation parameter values.

Table 6.2 Folded Flexure Resonator Parameter Experimental Hypervolume Results

Folded Flexure Resonator					
Crossover %	70	80			90
Mutation %	1	12.5	1	5	1
S^U	9.999999999931E-10	9.9999999998691E-10	9.9999999999786E-10	9.9999999999942E-10	9.99999999996923E-10
S^M	9.999986091057E-10	9.9999919301565E-10	9.9999999655176E-10	9.9999939855980E-10	9.99999995962131E-10
S^L	9.999943161240E-10	9.9999599160580E-10	9.9999998344580E-10	9.9999701359046E-10	9.99999980707823E-10

*$(S^U S^M S^L)$ [10, $1e^{-5}$]

Using the default values set out in table 6.1, five sets of experiments were undertaken using different combinations of parameters for the SBX crossover and polynomial mutation probabilities. The design optimisation of a folded flexure resonator was used to evaluate each of the parameter combinations over 5 separate tests at an individual cost of 14400 functional evaluations. The parameters chosen for investigation were 70%, 80%, and 90% SBX crossover probability, and 1%, 5%, and 12.5% polynomial mutation probability.

The hypervolume is calculated for each of the combinations tested and are shown in table 6.2, for the upper S^U, lower S^L , and mean S^M values, with shaded values indicated the best result. Also included in figure 6.1.1 are two plots of the population sets for the 1% and 5% polynomial mutation combinations, both having an 80% crossover probability.

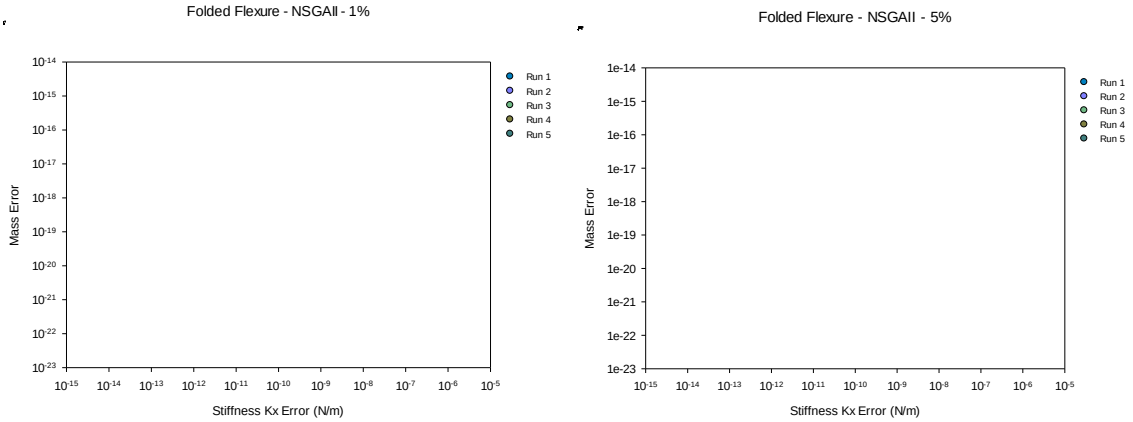


Figure 6.1 Folded flexure resonator final population sets for (left) 80% crossover / 1% mutation NSGAII and (right) 80% crossover / 5% mutation NSGAII

Looking over the hypervolume results for the combinations chosen it is clear that a value of 80% crossover is successful enough in providing optimal values for mean (S^M), upper (S^U) and lower (S^L) hypervolume. The mutation rate of 1% and 5% also contain such optimal values and therefore the choice of parameter probability for both SBX crossover and polynomial mutation rate should be one of these combinations.

Comparing plots of both sets of results for 1% and 5% polynomial mutation in figure 6.1 both provide good results for each objective, with the 1% mutation set providing population sets which are more compact and with less error on average compared with the 5% set. The best results based upon the objectives fall within the 5% sets, in particular runs 1 and 2, though run 2 for the 1% set is also comparable in quality. The default settings will have to cover a range of experiments, of which the previous bandpass system and device level ones are an example. The system level design optimisation utilised a rather high mutation rate, in what was an aim to aid topological search through mutation of the tank number. The choice of a higher mutation rate, in this case 5% seems applicable, though there is no guarantee this is the best value over the possible MEMS design problems to be outlined later.

Table 6.3 Variation Operator Parameter Values

Algorithm Parameter	Default Value
SBX Crossover Probability	80%
Polynomial Mutation Probability	5%

6.2 MEMS component design

MEMS are a multidisciplinary field, often composed of a number of individual components which together form a higher function or purpose. Outlined previously, a parallelized tunable filter consists of multiple micromechanical filters, each of which can consist of smaller units, be it micromechanical resonators or coupling spring components.

It is important to the design of the whole system to naturally be able to perform design synthesis and optimisation of each individual component within such a complex device as a tunable filter. The generic goals of cost (computational / time) and performance are what often drive such design synthesis and optimisation and it is these criteria which also drive the MEMS design problems to be outlined.

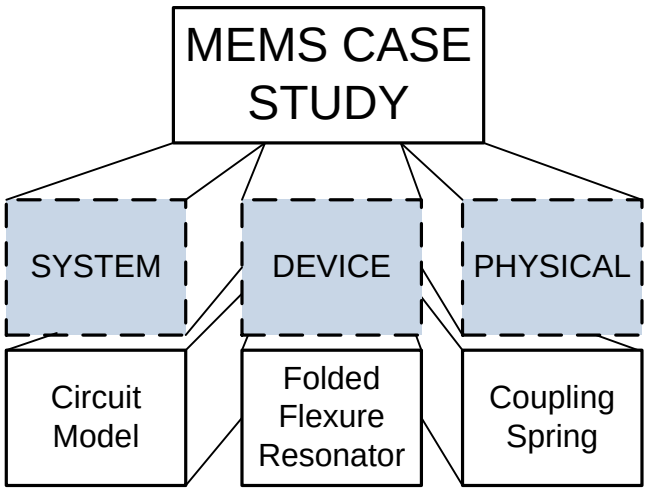


Figure 6.2 MEMS case study hierarchical levels and linked models and components

Outlined previously were three hierarchical levels of interest in MEMS design that is under investigation. The bandpass filter has also been outlined as the case study to be used and a validation of its design synthesis and optimisation has been undertaken in chapter 5. The decomposition of the bandpass filter over the various levels of design is a choice ultimately by the designer or design team, though constrained by a number of factors for example the modelling and analysis tools available. Outlined in figure 6.2 are the three individual levels utilized in the design synthesis of the bandpass filter and the components or models in which they cover, these being the electrical equivalent circuit model, the micromechanical folded flexure resonator component and finally the coupling spring component.

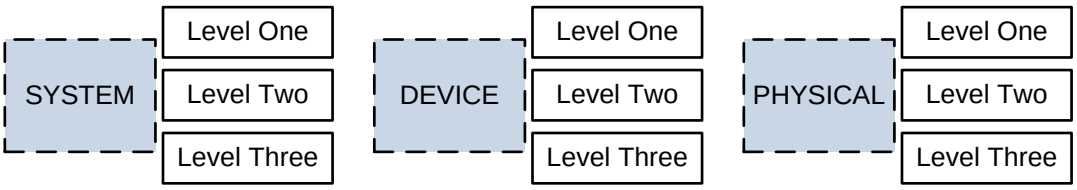


Figure 6.3 Uni level design optimisation of MEMS using multi-level design strategies

Each of these levels and their associated models or components can be utilized in the design optimisation of a MEMS bandpass filter at some stage of the design process. The application of multi-level design strategies can also be applied within a single level of MEMS modelling and abstraction with the aim to aid design synthesis and optimisation. As shown in figure 6.3, three of the levels have each been broken down into a number of levels depending on the multi-level strategy used and all form a series of experimental investigations into the applicability of such strategies at a single or 'uni' level of modelling and analysis.

6.3 System level design optimisation

Design synthesis and optimisation of MEMS at a system level often occurs either through the direct synthesis of circuit controlling and sensing elements at the board level [204] which are interfaced with the device itself, or by using system level abstract models be they simulink block diagrams [51] or electrical equivalent circuit models [55].

The synthesis of micromechanical bandpass filters through the use of electrical equivalent circuit models forms the basis for a system level design optimisation problem. A new frequency range is chosen which lies within the previous examples optimised in chapter 5 and the values are shown in table 6.5. The change in frequency range also means a change in variable parameter bound values which were altered to reflect the new frequency range and these are shown in table 6.4.

Table 6.4 System Level Filter Problem Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Voltage	Real Valued	1	200
Tank Number	Integer	1	9
Finger Number	Integer	1	200
Thickness (μm)	Real Valued	2	30
Capacitance (F)	Real Valued	3E-15	8E-15
Inductance (H)	Real Valued	40000	80000
Coupling Spring Capacitance (F)	Real Valued	3E-15	8E-15
Tank	Branch	N/A	N/A
Objectives		Constraints	
Bandpass Filter Response Error	Minimize	N/A	
Bandpass Central Frequency Error	Minimize		

The objectives for design synthesis and optimisation remain the same as in the previous chapter, and the multi-objective evolutionary algorithms used for this design problem are NSGAII and SPEA2. The number of functional evaluations remains at 10,000.

Table 6.5 System Level Parameter Ranges for Bandpass Filter

Bandpass Parameter	System Level
Passband	9.5kHz – 10.5kHz
Stopband 1	1Hz – 9.5kHz
Stopband 2	10.5kHz – 15kHz
Central Frequency	10kHz

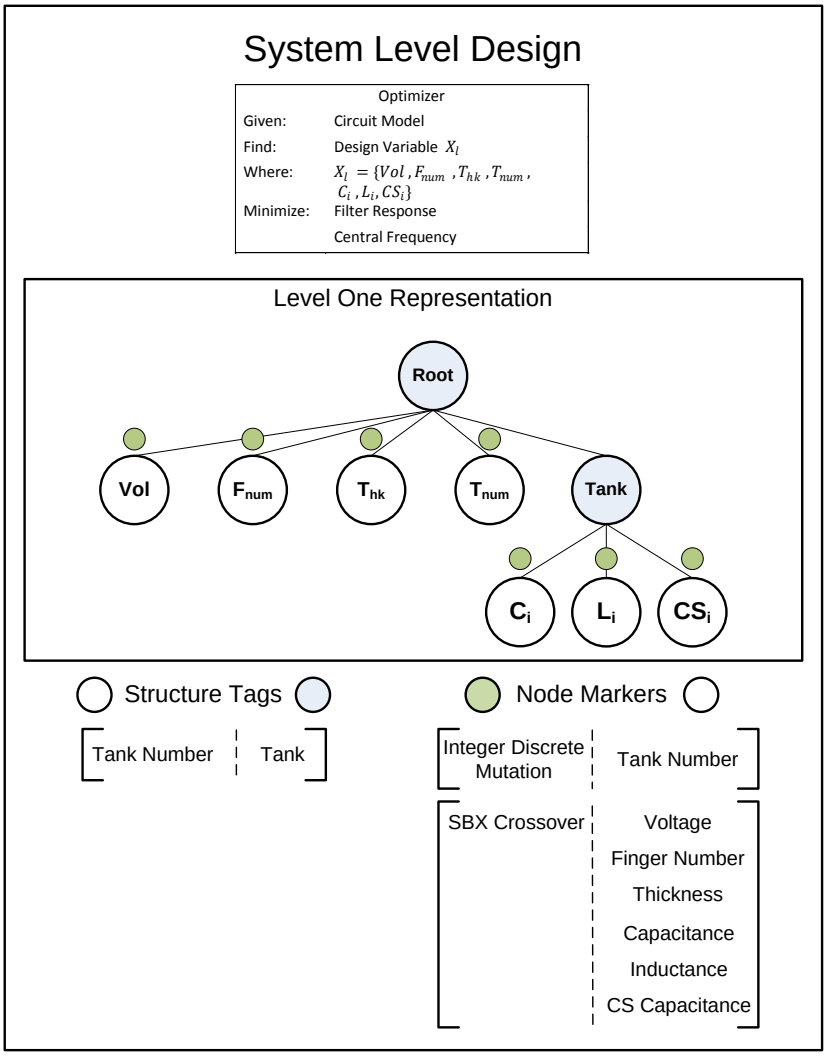


Figure 6.4 System level design template, with overview of problem, default representation, associated structure tags and node markers.

6.3.1 Numerical Results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.5 and fully in appendix C.1. Also shown are the best frequency transmissions for each algorithm, ranked by the filter frequency objective in figure 6.5 and fully in appendix C.1 along with their objective values in table 6.6. Finally the hypervolume values for both algorithms are shown in table 6.7, with the best results shaded.

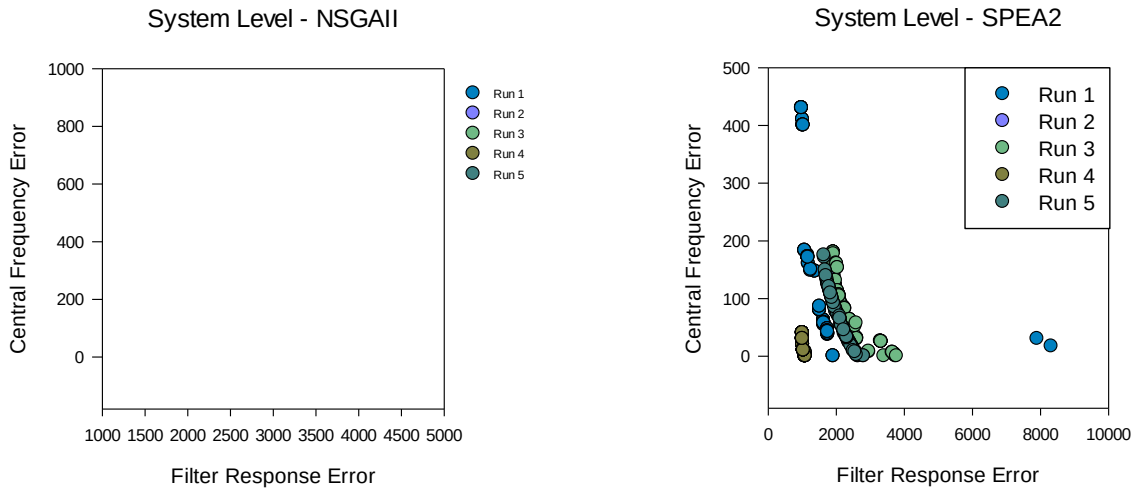


Figure 6.5 System level run 1 – 5 final population sets for (left) NSGAII and (right) SPEA2

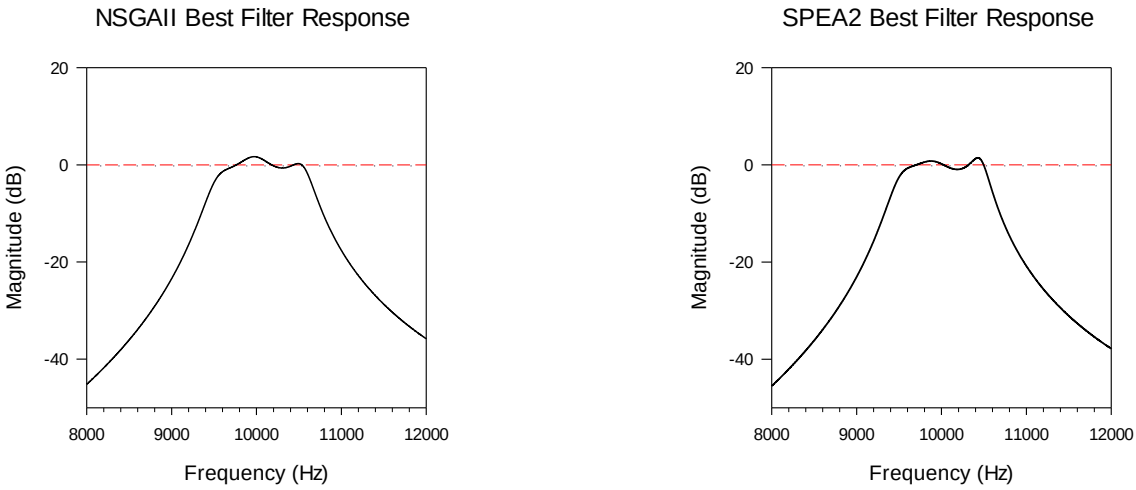


Figure 6.6 System level best filter response ranked by filter frequency objective for (left) NSGAII and (right) SPEA2

Table 6.6 System Level Bandpass Filter Results

System Level NSGAII					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	0	1750.493	101	61.34	3
2	0	1680.625	235	105.75	3
3	31	1248.642	32	190.34	3
4	0	3315.054	910	30.64	2
5	0	2148.439	206	144.01	3
System Level SPEA2					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	29	984.904	430	11.785	3
2	15	1936.521	160	199.10	3
3	1	1925.665	180	139.56	3
4	8	1012.157	40	85.27	3
5	80	1643.993	175	48.73	3

Table 6.7 System Level Hypervolume Results for NSGAII and SPEA2

System Level		
Hypervolume	NSGAII	SPEA2
S^U	43755994.2233912	44945558.0890751
S^M	39634066.167	42433896.104
S^L	32507262.7638433	40276362.6859817

* (S^U S^M S^L) [10000, 5000]

The single level design optimisation strategy stands as the current state of the art in automated stochastic design of MEMS and it is the base level at which all other strategies are compared. Both the single level NSGAII and SPEA2 were successful in finding reasonable bandpass transmissions, in particular runs 1 and 4 for the MOEA SPEA2. The strength of SPEA2 over NSGAII is mirrored in the hypervolume results in table 6.7 with superior results for upper, lower and mean values. In both sets of experiments there are examples of unacceptable bandpass transmissions, often for those solutions with filter frequency objectives greater than 1700.

Convergence to a front seems to be a common theme in both sets of results, as is the number of tanks used by the optimal solutions. A particular phenotype is evolved and then this forms the basis for further exploration along both objectives as seen in figure 6.7. Here the frequency transmission shape is structurally the same with only small variation in passband width and ripple shape over the three solutions. Convergence however can also be a problem if it falls upon a sub optimal area of the design search space, as seen in run four of the NSGAII population sets in appendix C.1. Even with the global search characteristics of NSGAII, this particular phenotype will probably not be able to evolve into better transmission shape characteristics seen in other runs simply because small changes in the transmission shape can have large effects on objective values as seen in figure 6.7. The required topological change from two to three resonator tanks, coupled with the individual *LCR* value changes are probably to great in a single bound, or incrementally, to reach a phenotype similar to those found by other runs.

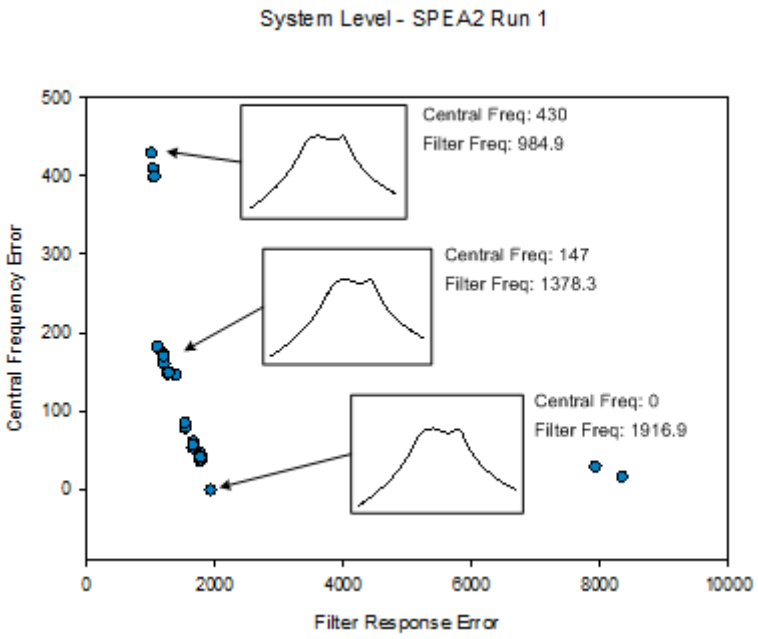


Figure 6.7 Phenotype spread over SPEA2 run one final population set

6.3.2 Multi-level evaluation

MEMS encompass a number of hierarchical levels in modelling and analysis, often connected to a unique set of tools or software. The system level, an abstract level of modelling contains tools such as simulink or circuit simulators like Spice which model block diagram or electrical equivalent circuit models.

The design strategy multi-level evaluation can be classed with the task of employing multiple levels of modelling or analysis, often with different levels of accuracy and cost, and often employed with the aim of reducing the computational cost of design optimisation. This can be emulated in MEMS design through the system, device and physical design levels outlined in previous chapters, or within a single MEMS design level itself.

The modelling and analysis of the bandpass filter involves the electrical equivalent circuit model and analysis through the circuit simulator Spice. The analysis involves a number of parameters which are directly linked to the accuracy and cost or speed of the analysis, these being the number of sampling points and the sampling range. Altering the sampling number and range therefore allows control on how fast and accurate an analysis is, essentially a trade off between the two and the ability to perform multi-level evaluation.

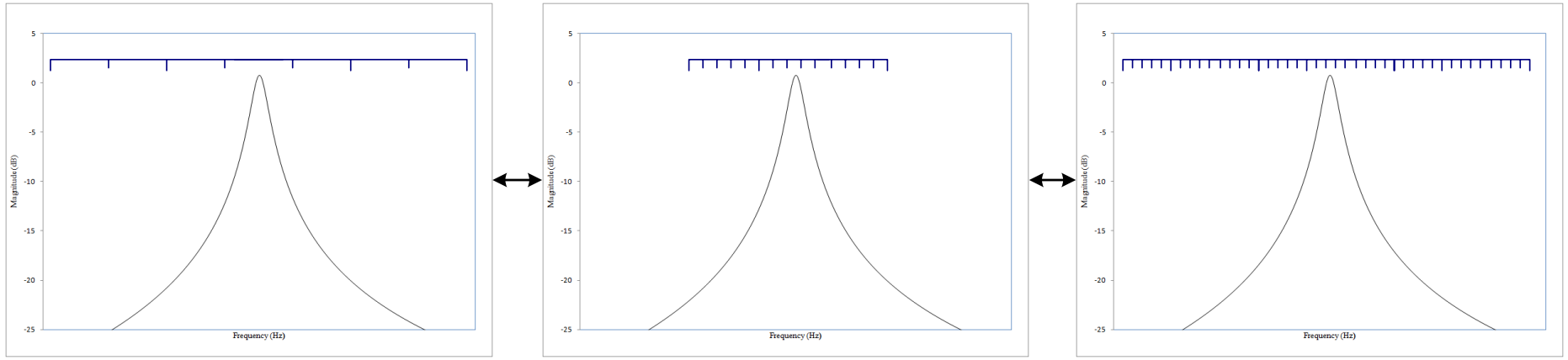
The number of functional evaluations that occur over the whole design process in evolutionary based algorithms plays an important role in the effectiveness they have in producing novel and optimal solutions. In a standard EA each functional evaluation is an outcome of any newly evolved solution and with constraint information provides information on the quality of the new design. The more functional evaluations that can occur the better, as it allows for further search within the design space.

Table 6.8 outlines three levels of evaluation within the circuit simulator Spice for analysis of the bandpass circuit model. The two lower levels, levels 1 and 2 provide a basic level of analysis of the frequency transmission at a third of the cost in comparison with the standard analysis employed in the previous experiment and used in level 3.

Table 6.8 Multi-Level Evaluation Circuit Model Analysis Parameters

	Level 1	Level 2	Level 3
Passband	9.5kHz – 10.5kHz	9.5kHz – 10.5kHz	9.5kHz – 10.5kHz
Stopband 1	1Hz – 9.5kHz	7.5kHz – 9.5kHz	1Hz – 9.5kHz
Stopband 2	10.5kHz – 15kHz	10.5kHz – 12.5kHz	10.5kHz – 15kHz
Sampling Size	5000	5000	15000
Central Frequency	10kHz	10kHz	10kHz

Each of the three levels are visualised in figure 6.8, here level 1 covers the same range as the original analysis but only samples a third of the frequency range with 5000 points, while level 2 employ a different strategy of a restricted frequency window with a focus on the bandpass region of interest. The final level cover the entire frequency range of 1 Hz to 15 kHz and samples each of the 15,000 frequency points.



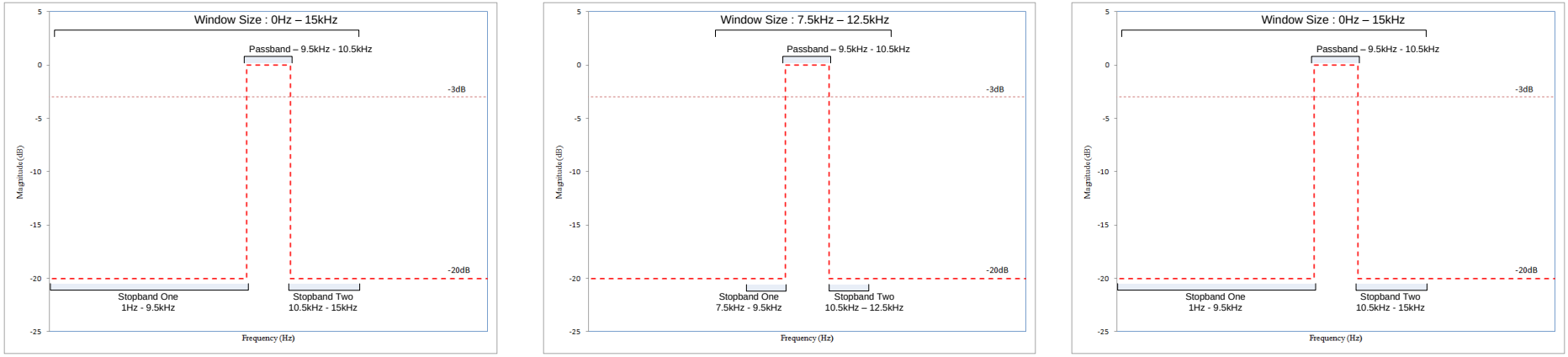


Figure 6.8 Multi-level evaluation circuit model analysis sampling size and range characteristics - (top) sampling size values of 5000 (level 1) and (level 2) and 15000 (level 3) frequency points (bottom) sampling frequency ranges with full ranges 1Hz – 15kHz (level 1) and (level 3) and reduced range 7.5kHz – 15kHz (Level 2)

The design process employed is a population based multi-level optimisation, where three separate and isolated evolutionary algorithms are structured to evolve their own population of solutions over a fixed period of generations or functional evaluations. Each of the separate levels employs the same default algorithmic parameters outlined in table 6.1 and 6.3, while the representation remains unchanged from the standard bandpass problem in figure 6.4. The multi-level evaluation design problem is outlined in figure 6.9, with each level communicating hierarchically between its neighbours.

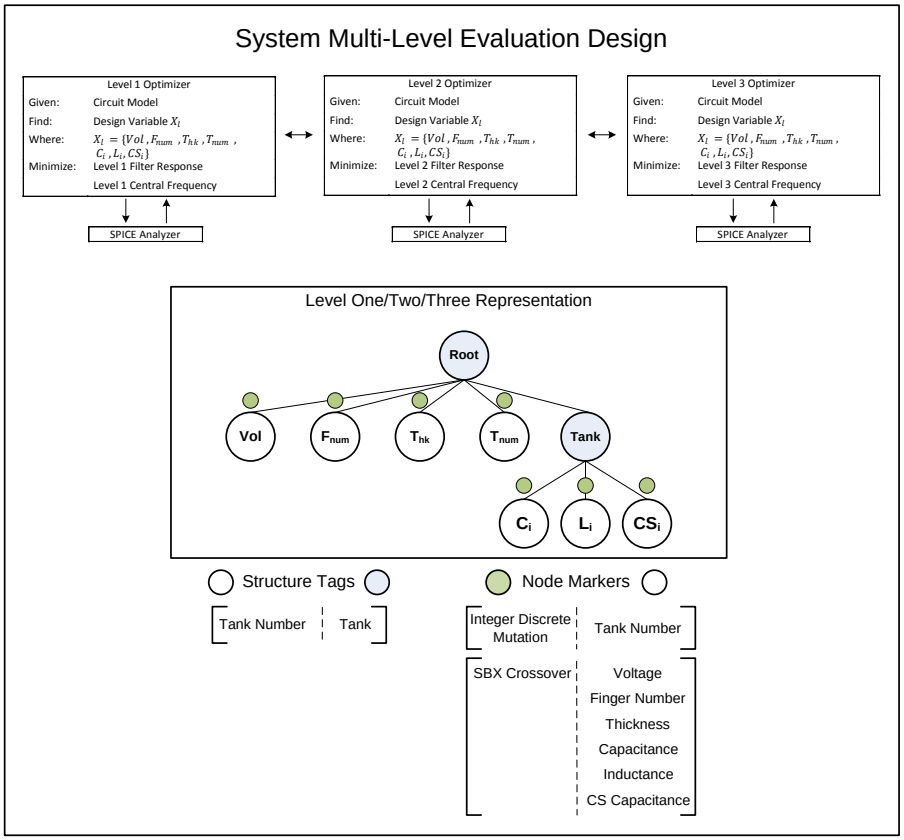


Figure 6.9 System multi-level evaluation design template, with overview of problem, default representation, associated structure tags and node markers.

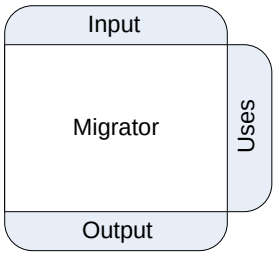


Figure 6.10 Migrator framework module

In order to transfer solutions from one level to another a new migrator framework module is required. Shown in figure 6.10 the migrator module is designed to handle the transfer of individual or sets of solutions from one population set to another. The migrator module contains a migration percentage parameter to indicate the number of individuals to transfer from one population set to another. The selection of individuals is based upon the same process within

the standard NSGAII or SPEA2 algorithms, for example Pareto ranking and crowding, selecting the chosen percentage of individuals similar to truncation. If necessary as is the case here the chosen solutions are re-evaluated at the level specific analysis and then given a level specific rank and crowding value. A combination of both population sets then occurs similar to the standard Pareto ranking and crowding replacement to give the final population set. Successful migrants therefore remain within the final population while those of worse rank do not.

Four migrator modules are utilized in this particular design strategy to allow individuals to move between neighbouring levels. The values for migration percentage along with the cycle count when migration is invoked are shown in table 6.9. The overall multi-level evaluation pathway for system level design optimisation is shown in figure 6.11, each level is outlined and involves the modules that make up an individual MOEA in this instance NSGAII.

Table 6.9 Migrator Module Parameters for Multi-Level Evaluation

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	20	4
Level 2	Level 1	20	4
Level 2	Level 3	20	4
Level 4	Level 3	20	4

The objectives for design synthesis and optimisation remain the same as in the previous chapter, and the multi-objective evolutionary algorithms used for this design problem are NSGAII and SPEA2. The number of functional evaluations remains at 10,000, however the cost of each evaluation is different depending on the level it is undertaken.

Table 6.10 Evaluation cost for SPICE electrical equivalent model

Full SPICE Analysis		Reduced SPICE Analysis	
Frequency Range	Sampling Size	Frequency Range	Sampling Size
1Hz – 15kHz	15,000	1Hz – 15kHz	5,000
Mean Analysis Time (20 Calls)		Mean Analysis Time (20 Calls)	
0.769344005		0.41932398	
Ratio			
1 : 1.834724561			

Above is a table of the required time to undertake analysis of a standard 5 tank electrical equivalent circuit model for a bandpass filter in the tool Spice. Two sets of analysis were undertaken, the standard analysis for this particular problem of 15,000 sampling points over the range 1Hz – 15kHz, and finally over the same range but only sampling 5000 points. The ratio between both means gives a near 1:2 ratio in favour of the smaller sampling set. As a result each functional analysis call to the SPICE simulator at levels 1 and 2 have a cost of 0.5 in relation to the full cost of 1 functional evaluation at level 3. The total global functional evaluations of 10,000 are a constraint on the design process and the optimisation environment is stopped when this target is met. Each of the levels is called sequentially from level one to level three, with levels one and two called at every cycle while level 3 is called twice in every 4 cycles. Giving the outlined cost ratios this equated to about 3300 functional evaluations for levels one and

two, and 3300 functional evaluations at level three, though the real number is slightly lower due to functional evaluation costs occurred during migrations between levels.

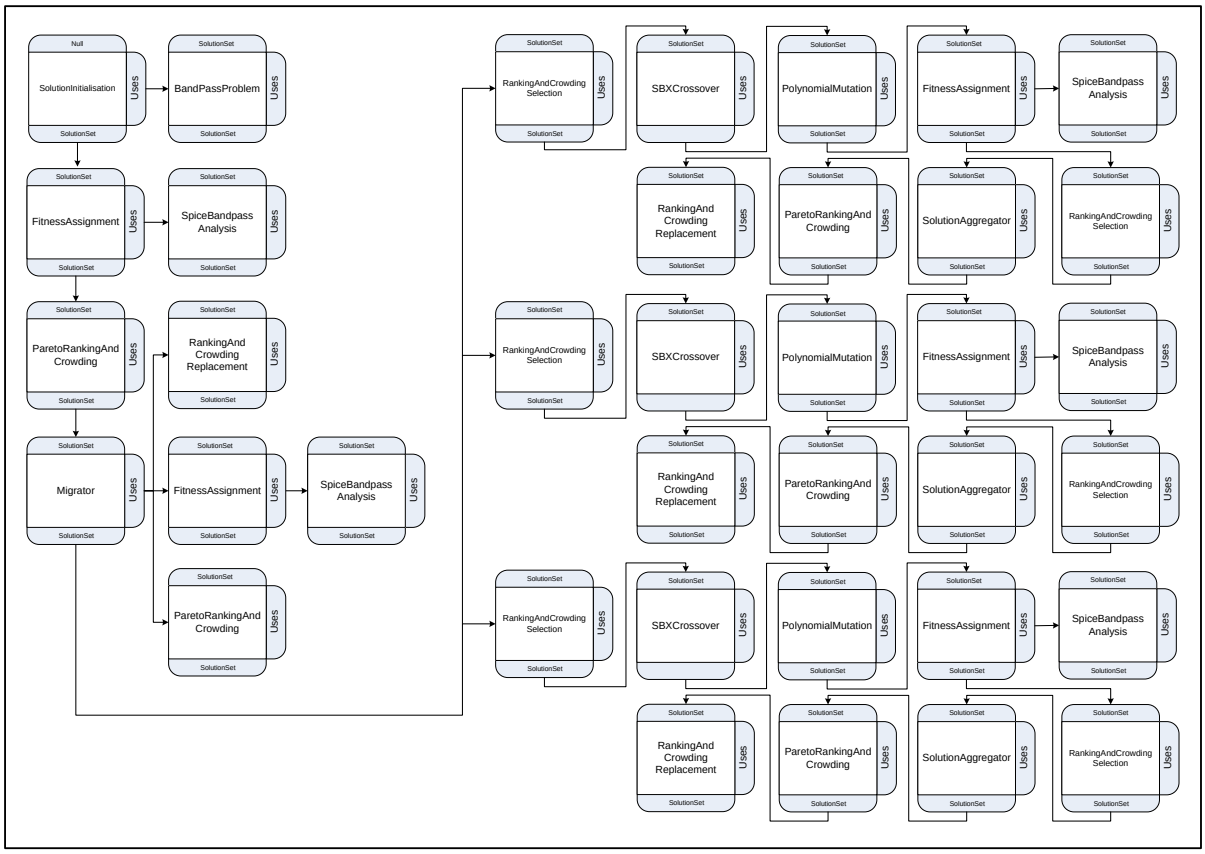


Figure 6.11 System multi-level evaluation NSGAII environment pathway

6.3.3 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.12 and fully in appendix C.1. Also shown are the best frequency transmissions for each algorithm, ranked by the filter frequency objective in figure 6.13 and fully in appendix C.1 along with their objective values in table 6.11. Finally the hypervolume values for both algorithms are shown in table 6.12, with the best results shaded.

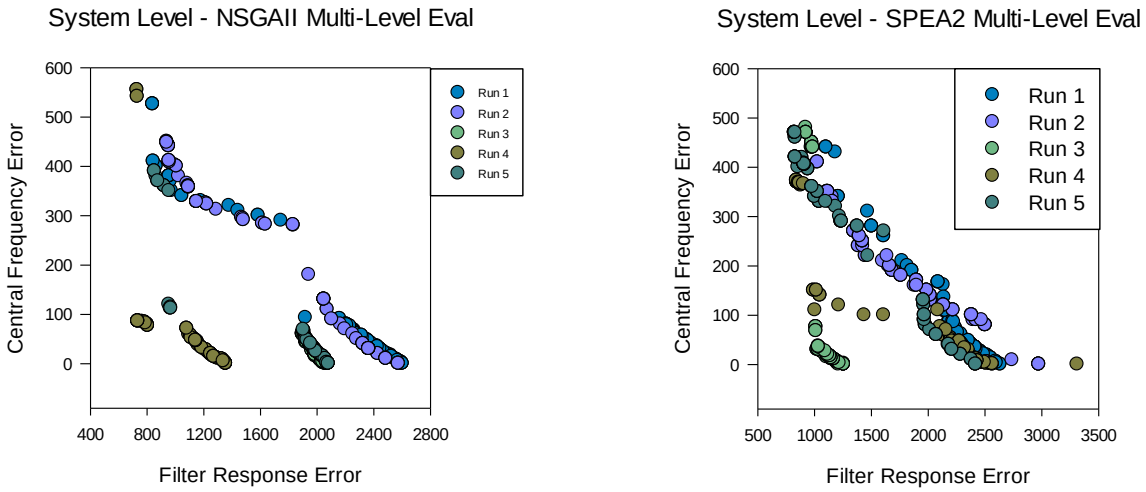


Figure 6.12 System level run 1 – 5 final population sets for (left) NSGAII multi-level evaluation and (right) SPEA2 multi-level evaluation

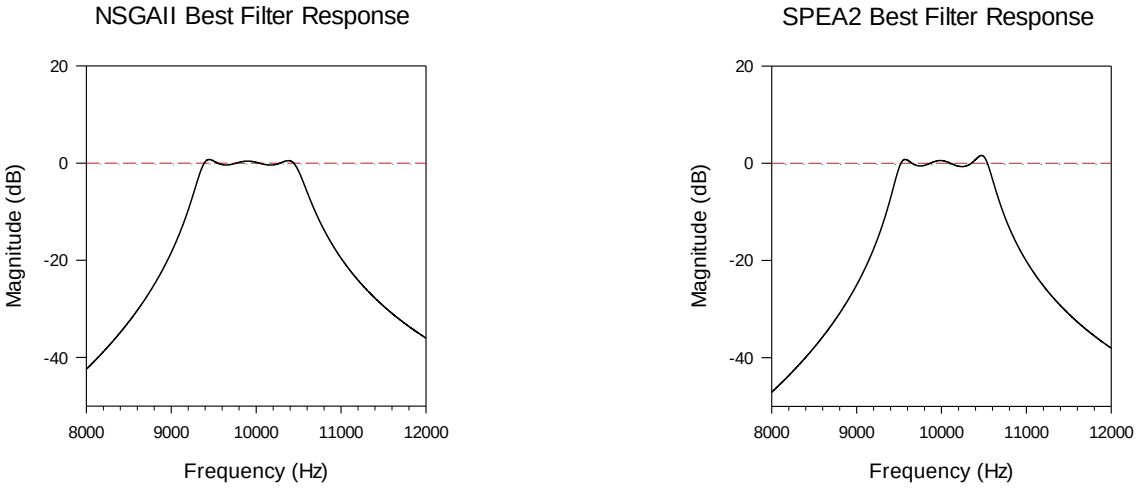


Figure 6.13 System level best filter response ranked by filter frequency objective for (left) NSGAII multi-level evaluation and (right) SPEA2 multi-level evaluation

Table 6.11 System Multi-Level Evaluation Bandpass Filter Results

System Multi-Level Evaluation NSGAII					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	1	841.425	526	22.99	3
2	1	940.822	450	129.56	3
3	5	1897.78	60	31.145	3
4	0	730.915	555	12.57	3
5	82	853.477	390	60.44	3
System Multi-Level Evaluation SPEA2					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	30	1105.505	440	125.97	3
2	0	1028.340	410	99.08	3
3	22	924.687	480	155.89	3
4	4	844.511	373	9.59	3
5	8	826.304	470	179.27	3

Table 6.12 System Level Hypervolume Results for NSGAII and SPEA2 Multi-Level Evaluation

System Multi-Level Evaluation		
Hypervolume	NSGAII	SPEA2
S^U	46307464.8585	45592440.8746
S^M	44545442.897	45033152.979
S^L	40506807.6571	44162948.0661

* (S^U S^M S^L) [10000, 5000]

Multi-level evaluation as a strategy for the design optimisation of a bandpass filter can be seen to provide a better set of solutions over the two different algorithms than their single level counterparts in section 6.3. Both the frequency transmissions in figure 6.13 and appendix C.1, and their objective data in table 6.11 show a robust set of results with superior objective values over the ten separate runs when compared with the single level approach. Both NSGAII and SPEA2 provide similar performance, however with regards to NSGAII one particular run failed to match the characteristics of all the others. This is reflected in the hypervolume results in table 6.12, where even though NSGAII found the best set of results in a single run, the mean performance is dragged down by run 3, the worst hypervolume value. The results for SPEA2 also demonstrate the effect increased rolloff has on the filter objective value, running through each of the runs 1 to 5, the increased sharpness mirrored in the filter objective values.

There are a number of factors which could contribute to this increase in performance, these include:

- Additional population sets
- Increase in number of functional evaluations
- Variation in design search space

The addition of demes into the design process allows for the possibility of increased diversity [170] as the semi-isolation of each population set allows for each one to evolve along different evolutionary lines. The migrator module allows for good solutions to communicate between the population sets and as a result over time each population set will probably begin to resemble one another as they all converge to a particular pareto front.

The diversity of the population and its convergence to a front over the design process play an important role in the overall outcome of the final population set and the solutions within them. Commonly this is attributed to the two dynamic and often conflicting paradigms often coined as ‘exploration’ and ‘exploitation’ which themselves are affected by the overall selection pressure of the design process.

Looking over the single level results for bandpass filter design, the majority of the final populations have converged neatly to a particular Pareto front, this is more evident in figure 6.5 for the NSGAII final population sets. This particular region of the objective space, characterised generally as a filter response error range of 2k to 3k and central frequency error of 1Hz to 300Hz, is most evident in the runs 1, 2 and 5 of NSGAII and 2, 3, and 4 of SPEA2, of figure 6.5. The phenotype responses shown in appendix figures C.3 and C.4 for each of these runs are also very similar.

This particular region is also populated in some of the multi-level evaluation final populations, in particular runs 1, 2, 3, and 5 for NSGAII multi-level evaluation and runs 1, 4, and 5 for SPEA2. However rather than finding themselves restricted to this particular region, both the NSGAII and SPEA2 multi-level evaluation approaches were able to explore other areas of the objective space finding better filter response solutions.

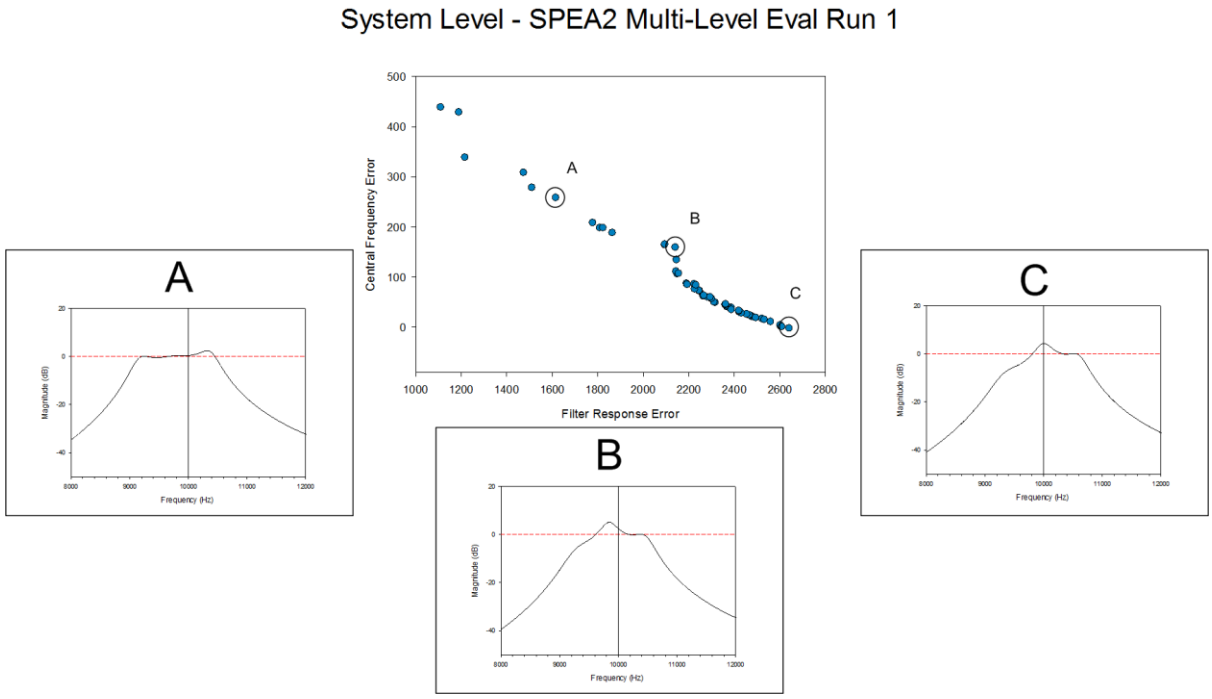


Figure 6.14 Final population plot for system multi-level evaluation run 1 with corresponding filter transmissions for solutions A, B and C.

Figure 6.14 shows an example of a final population set for a SPEA2 multi-level evaluation run, along with some of the filter transmissions evolved. Here within the region outlined previously two solutions are chosen, B, and C, showing the transition between the filter transmission shapes. The transition between both filter shapes and the solutions in between are a result of the central peak in solution C shifting away to give a broader, flatter passband in solution B, a solution which is typical of the filter transmissions found by the single level strategies. This is by far the farthest the majority of the single level design strategies go in terms of performance.

The multi-level design strategies however seem more able to break away from this region as shown in solution A, and ultimately the best solution found by SPEA2 multi-level evaluation run 1 in appendix figure C.8. The filter transmission is beginning to resemble a bandpass filter as solutions are evolved with flatter passbands. The only example where the multi-level strategy failed was run 3 for the NSGAII algorithm, where interestingly it converged to an identical front to that found in run 5, however in this particular run the algorithm was able to explore further, showing it is possible to break out of this particular region of the objective space. The difficulty to do so could be as a result of what looks to be a discontinuous search space, shown in runs 1, 2, and 5 for NSGAII and 1, 4, and 5 for SPEA2 in figure 6.12.

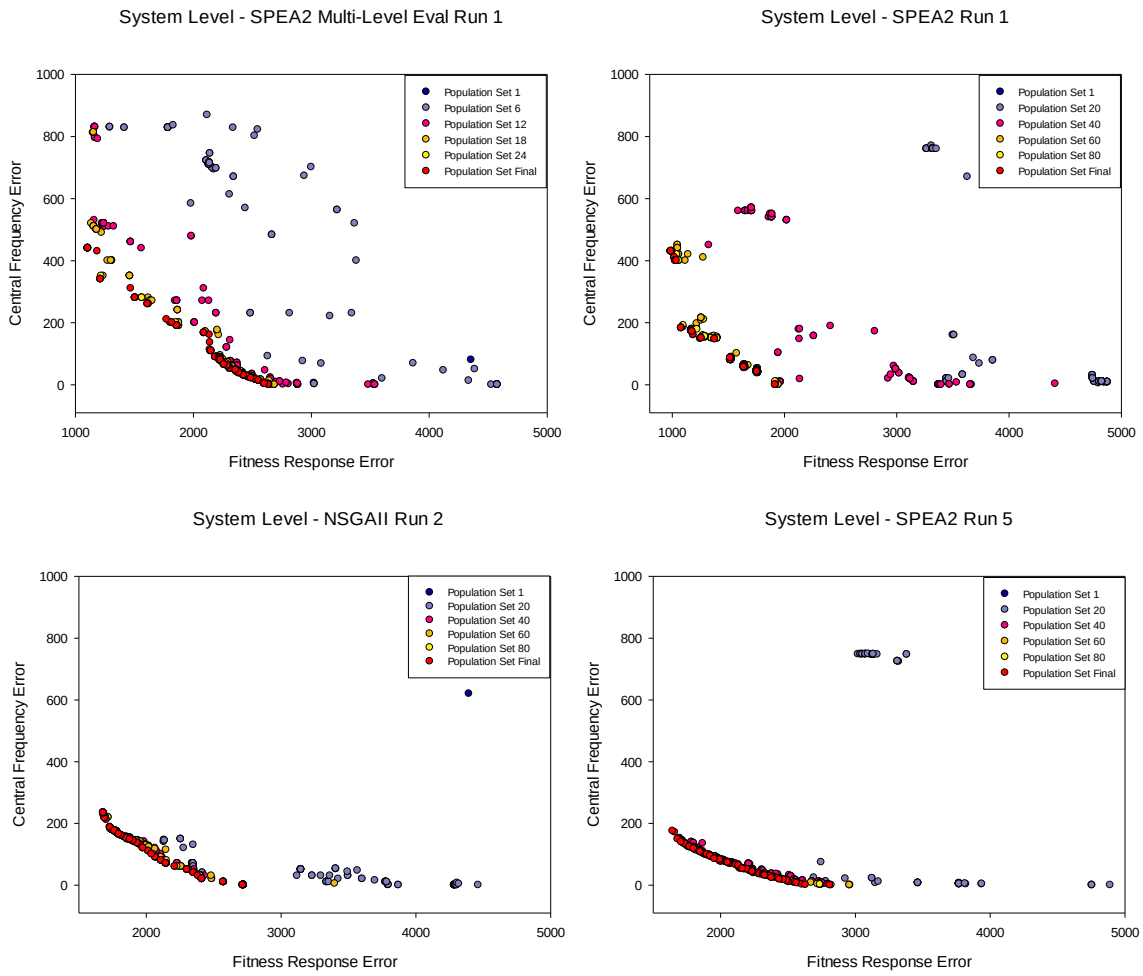


Figure 6.15 Generational population plots for both single and multi-level runs – each plot contains 6 equally distant generational plots

An analysis of a number of generational population sets for both single and multi-level system design in figure 6.15 highlights the failures of early convergence and how successful results contain more diversity with regards to the objective space. In this figure both SPEA2 single level run 1 and multi-level run 1 provided final population sets of equal quality, and both demonstrate the maintenance of diversity from early on, with solutions from the third population set lying within the 1000 and 1500 filter frequency error region of the objective space. Below this are two examples of convergence and overall poor quality transmissions, here the populations quickly converge to the final pareto front, with little diversity in any of the generational population sets.

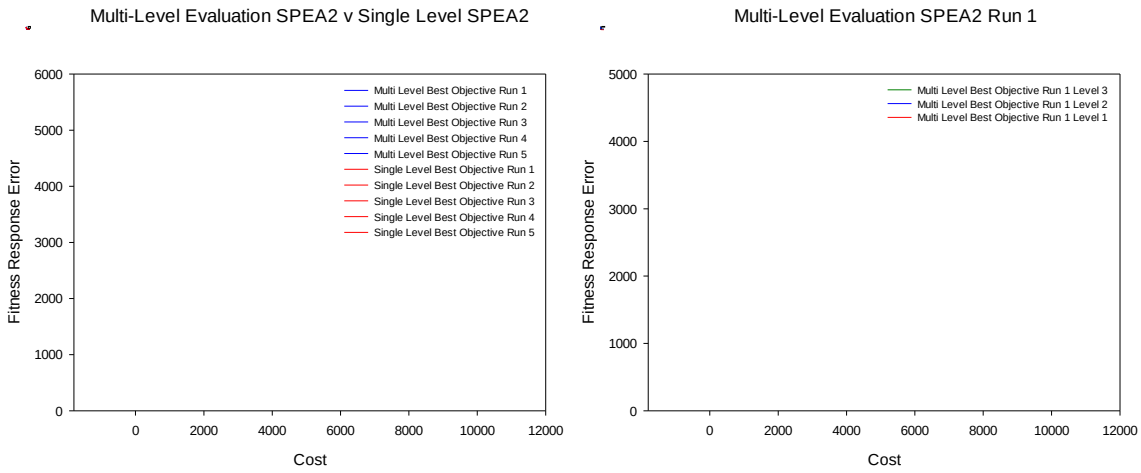


Figure 6.16 Best filter response objective values vs. cost for both multi-level evaluation and single level SPEA2 runs (Left) and SPEA2 multi-level evaluation run 1 plots for levels 1, 2 and 3 (Right)

The dynamic of 'exploration' and 'exploitation', in particular with regards to convergence, seems given the results and analysis presented to play a considerable role in the final outcome of the design process. Figure 6.16 shows the generational evolution of the best solution ranked by filter response error against the functional cost for both single level SPEA2 and multi-level evaluation SPEA2 runs. The majority of filter response error reduction occurs early on in the design process, with a clear view of the multi-level strategy outperforming the single level method which more often than not plateaus with little improvement. Also shown in figure 6.16 is a breakdown of the same best filter objective values for each of the three levels employed in run 1 of the multi-level evaluation strategy. The vertical grey lines indicate some examples when a migration of solutions occurs between all the levels and it can be seen the clear effect it has on the most accurate level 3, with better filter response solutions migrating to the population from level 2.

Interestingly the majority of the gain in performance occurs before the additional functional evaluations gained through the multi-level strategy begin. In figure 6.16 the cost is related to the cpu time and the analysis call, therefore a cost of 10,000 is equivalent to 10,000 functional calls at the single level, while for multi-level evaluation the ratio between cost and calls is no longer 1:1. This equates to about 6000 cost at around 10,000 functional calls, while the total number of calls in a single multi-level run is around 16500. Figure 6.17 shows the average hypervolume values for the single and multi-level evaluation strategies of SPEA2 and NSGAII against the cost of functional analysis. The superiority of the multi-level strategy over the single level one is once again clear in both algorithms, with both showing similar characteristics at an early stage of the design process, before the hypervolume values begin to plateau and show little improvement. However the single level approach begins to stall and converge much earlier, retarding performance.

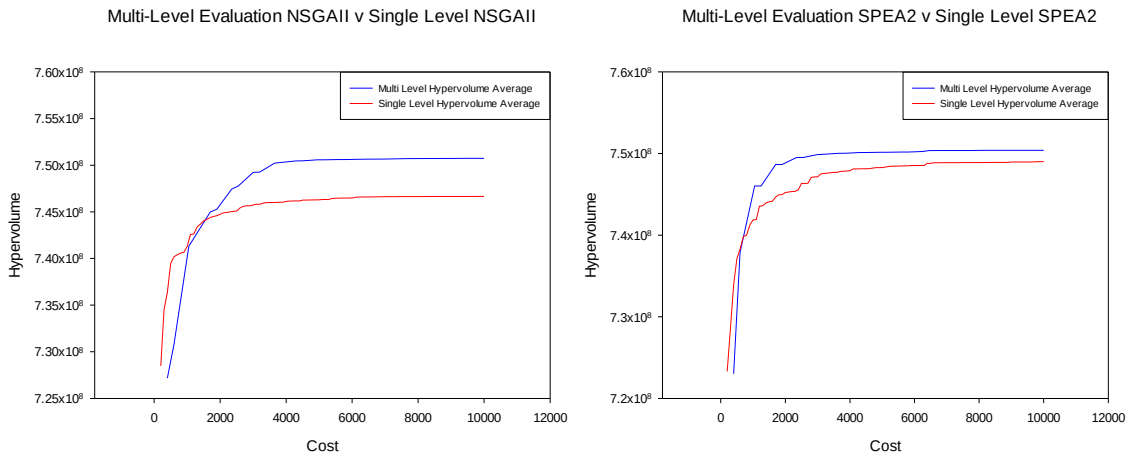


Figure 6.17 System level average hypervolume results for the 5 runs of the multi-Level evaluation and single level NSGAII and SPEA2 strategies * ($S^U S^M S^L$) [182000, 4150]

6.3.4 Multi-level parameterization

Design synthesis and optimisation within the field of engineering is often complex, in part due to devices which contain many design variables, non-linear behaviours and can be highly constrained. This complexity is often reflected in the design search space, which can also be highly multi-modal, discontinuous and constrained. Reducing the complexity might alleviate some of the burden placed on the optimisation algorithm employed to synthesize the device. The number of design variables present within the design problem effect the decision space the optimizer acts upon, and assuming all things are equal, the more design variables within a particular problem, the more engagement the optimisation algorithm may have to act on this decision space. Lowering the number of variables, or perhaps their bounds, that an optimizer can act upon may also reduce the complexity of the decision space it acts upon.

The multi-level parameterization strategy for bandpass design synthesis follows this tract by utilizing a number of levels each containing representations which have various design variables active within them. The design synthesis of a MEMS bandpass filter involves the coupling of multiple folded flexure resonator devices together with soft coupling structures. The frequency signal provided as input into the comb transducer component is then passed through this structure mechanically and transformed back as a filtered response.

The electrical equivalent circuit model includes these elements through the '*LCR*' resonator tanks and variables for the comb transducer and with an increasing number of tanks comes an increasing number of variables for which the optimizer has to evolve towards optimality. It is possible to simplify this by removing control of certain design variables and in the case of the resonator tanks, clone the values that make up the '*LCR*' resonator tanks so only one tank is evolved even within a multiple resonator solution.

The system multi-level parameterization design problem is shown in figure 6.18 and consists of two levels each with a different representation. The objectives and analysis remain the same as the default system level problem in section 6.3, as are the algorithm parameters for NSGAII and SPEA2. An additional node marker 'Level One' is used to denote the variables which are active at that level and can be altered by the variation operators. A new inclusion to the representation is

the inclusion of global variable information, the ability to allow the control of multiple variables of the same type through a single node. This is achieved by adding global variables which contain a tag name of the node type to be controlled, and an index value, which represents the node which is to be global after a breadth first search indexes all variables of that particular tag name. In figure 6.18 the variable nodes inductance and coupling spring capacitance are chosen to be global, and for example the value for inductance from the first tank or index 1 overrides all over inductance values for any other tank. The effect of this global variable 'cloning' is shown in figure 6.19 with the leaf nodes for these two variables under direct control of their global nodes. This allows control of numerous tank components within the representation but with only having to alter two variables rather than additional ones from each tank. In addition to this the upper bounds on the number of possible tanks has been reduced to 3 for the level one representation as shown in table 6.13.

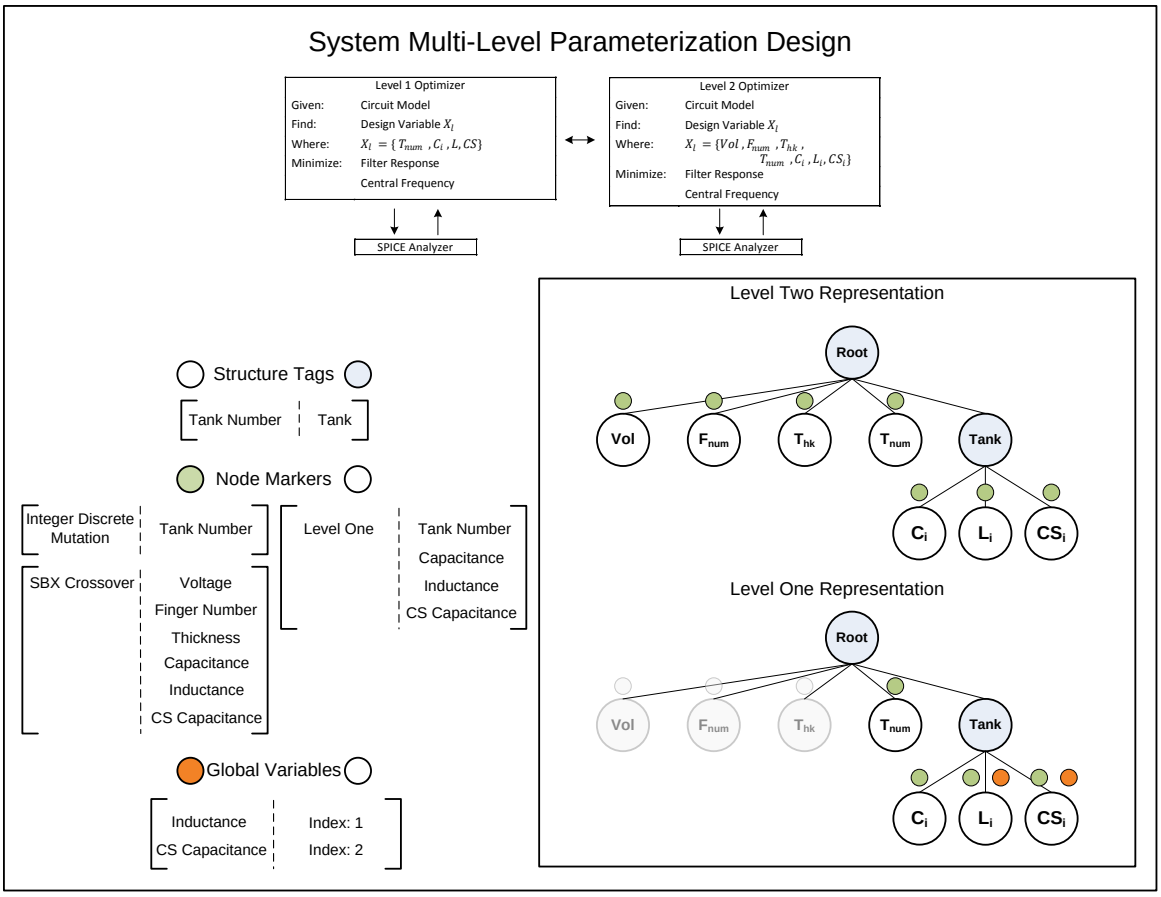


Figure 6.18 System multi-level parameterization design template, with overview of problem, default representation, associated structure tags, node markers and global variables.

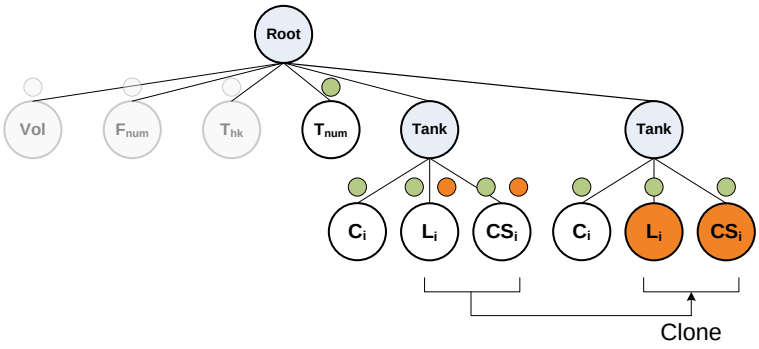


Figure 6.19 System multi-level parameterization global leaf variable clones

Table 6.13 System Multi-Level Parameterization Level One Filter Problem Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Tank Number	Integer	1	3

Table 6.14 Migrator Module Parameters for Multi-Level Parameterization

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	50	2
Level 2	Level 1	20	2

The migrator module only acts between the two levels, the parameters of which are shown in table 6.14, here the cycle count between each migration event has been reduced to allow more migration to occur, and the percentage increased for level 1. The module was also upgraded to allow transformation of solutions before migration, this was done to remove or add the global variables, listed in figure 6.18, or alter the tank number upper bounds from solutions depending on the level in which they are migrating too.

Each level has the same functional cost for each analysis, and therefore each levels design process is simply called sequentially, which gives a result of around 5000 functional generations for each level.

6.3.5 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.20 and fully in appendix C.1. Also shown are the best frequency transmissions for each algorithm, ranked by the filter frequency objective in figure 6.21 and fully in appendix C.1 along with their objective values in table 6.15. Finally the hypervolume values for both algorithms are shown in table 6.16, with the best results shaded.

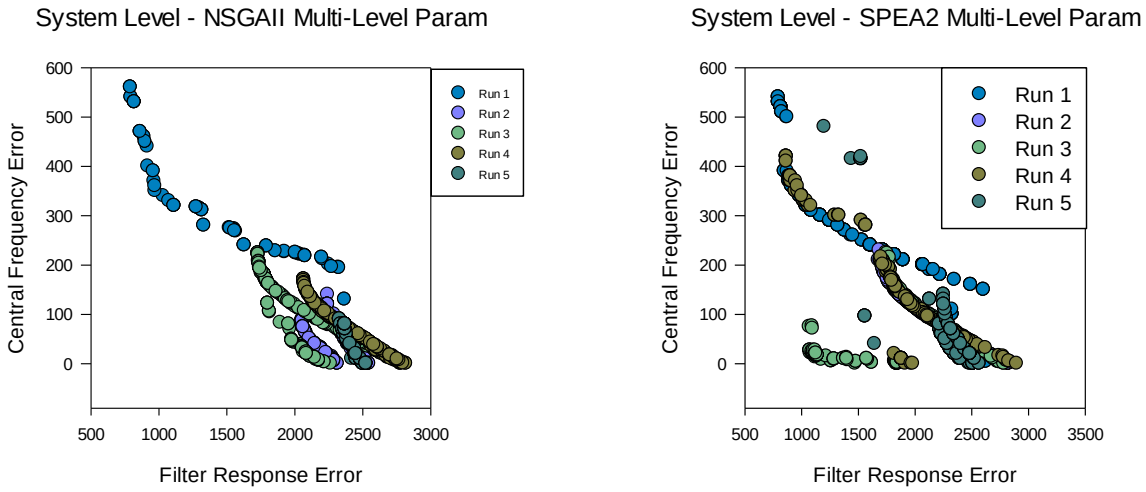


Figure 6.20 System level run 1 – 5 final population sets for (left) NSGAII multi-level parameterization and (right) SPEA2 multi-level parameterization

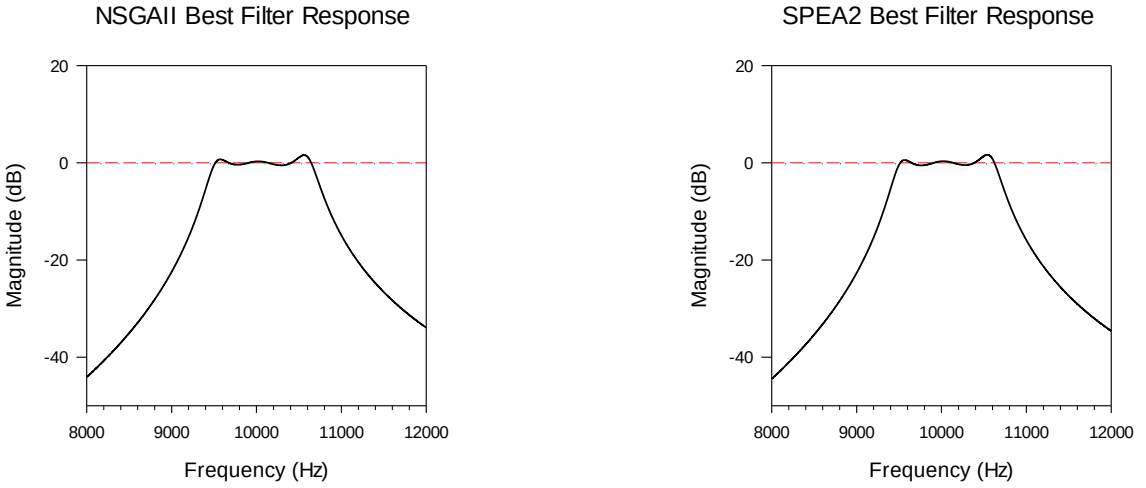


Figure 6.21 System level best filter response ranked by filter frequency objective for (left) NSGAII multi-level parameterization and (right) SPEA2 multi-level parameterization

Table 6.15 System Multi-Level Parameterization Bandpass Filter Results

System Multi-Level Parameterization NSGAII						
Test	Index	Level	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	0	Hi	792.742	560	119.26	3
2	39	Hi	2054.437	88	168.12	3
3	1	Hi	1730.289	224	157.85	3
4	1	Low	2067.765	171	187.82	3
5	5	Hi	2330.318	90	74.75	3
System Multi-Level Parameterization SPEA2						
Test	Index	Level	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	4	Hi	795.958	540	86.53	3
2	57	Hi	1678.407	210	73.47	3
3	7	Hi	1069.779	75	186.63	3
4	96	Low	866.147	410	47.11	3
5	10	Hi	1199.973	480	109.70	3

Table 6.16 System Level Hypervolume Results for NSGAII and SPEA2 Multi-Level Parameterization

System Multi-Level Parameterization		
Hypervolume	NSGAII	SPEA2
S^U	45579384.4953333	45595351.047224
S^M	40913176.976	43962322.184
S^L	38342249.202637	41513986.5053333

* (S^U S^M S^L) [10000, 5000]

The performance of the multi-level parameterization strategy is mixed over the two algorithms. The NSGAII approach overall has poor performance with regards to the frequency transmission for the bandpass filter failing on 4 out of 5 tests. On the other hand the multi-level parameterization SPEA2 approach shows improvement over its single level version and is comparable with the multi-level evaluation strategy though perhaps as less robust with run 2 seemingly stuck in a sub-optimal solution set and throwing off the mean hypervolume values associated with the algorithm.

Looking over at the final Pareto sets for each algorithm it is interesting to see that in a number of runs both levels have settled on different fronts of the search space. This can in some way be related to the use of multiple populations which allow for separate populations to evolve along different evolutionary lines, or perhaps on the granularity of the representation placing constraints at level 1, restricting search or leading to premature convergence.

The increased migration percentage and frequency probably discount the first assumption as each population is routinely updated with solutions from either level. Any solution drawn from level two to level one undergoes a transformation, altering the structure to have an upper bound of only 3 tanks, and application of global variables. As a result the frequency transmission of the circuit model can and most probably will change, and as a result alter the solutions objective values. This can make it harder for solutions to successfully migrate from level two to one, than vice versa.

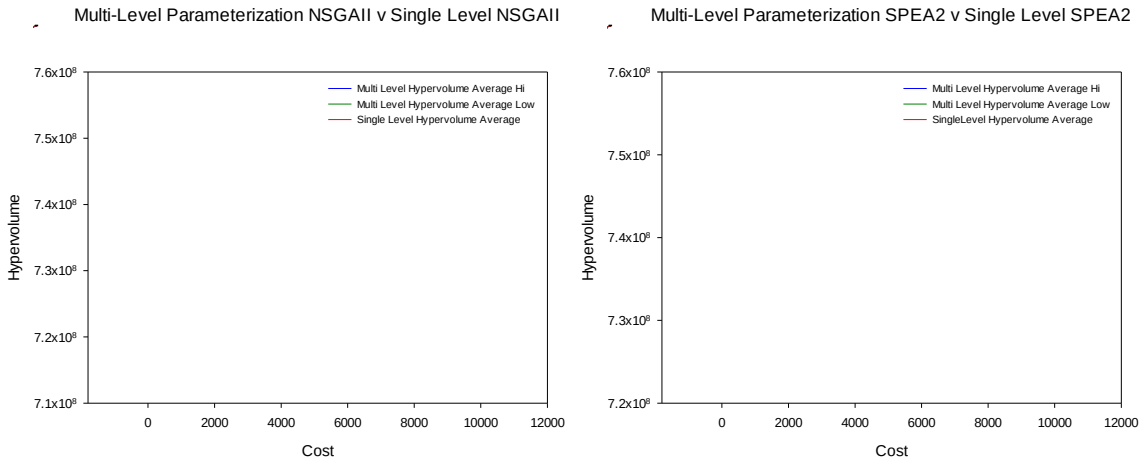


Figure 6.22 System level average hypervolume results for the 5 runs of the multi-Level parameterization and single level NSGAII and SPEA2 strategies * (S^U S^M S^L) [182000, 4150]

The generational hypervolume plots in figure 6.22 for both levels of NSGAII and SPEA2 show improved performance over the single level strategy, mainly through the highest level of parameterization. Figure 6.23 contains numerous experimental data for three separate runs, SPEA2 run 3 and 4, and NSGAII run 2. The results plotted are the best filter response objective value and the successful migration count against functional cost for both level two (Hi) and one (Low) population sets, and finally the generational population sets for both level one and level two.

The experimental run 3 of SPEA2 multi-level parameterization ended with two distinct pareto fronts, the more common front within the 2k and 3k fitness response error objective space, and finally the more optimal 1k to 2k front. Each of these fronts is associated with the final population set of one of the parameterized levels and this is seen through the generational plots in figure 6.23. Here the reduced parameterized level one is stuck at a sub optimal front and remains so throughout the whole design process. The unaltered representation level two final population set escapes this local optima around population set 36 (Cost 8000) onwards.

Separately run 4 of SPEA2 shows a close correlation in both the best filter response fitness and the generational population sets. The migration of solutions for both levels one and two consistently drops off over the design process, possibly as a result of convergence of each of the population fronts. It is interesting that both runs 3 and 4 low levels converge to the same sub optimal region of the objective space but run 4 is able to escape it early on with population set 9 and 18. Given the correlation between both the best filter response fitness and population sets, phenotypically the solutions must remain very close in order that any change as a result of migration from level two to one does not alter the objective values greatly.

This does not seem to be the case for run 2 of the NSGAII multi-level parameterization strategy; here structurally the generational population sets mirror one another for both levels one and two. However the filter response error is shifted backwards for the reduced parameterized population sets. The best filter response objective plots also indicate a correlation between both levels, with an improvement from one level feeding into the other.

Each of the three separate runs exhibit different behaviours and outcomes. Some contained levels which converged to sub optimal fronts while the other level was able to break free, others saw no distinction between them, with solutions moving freely between the objective spaces of either level relatively unaffected, while another sees a shift in objective space characteristics for their population sets.

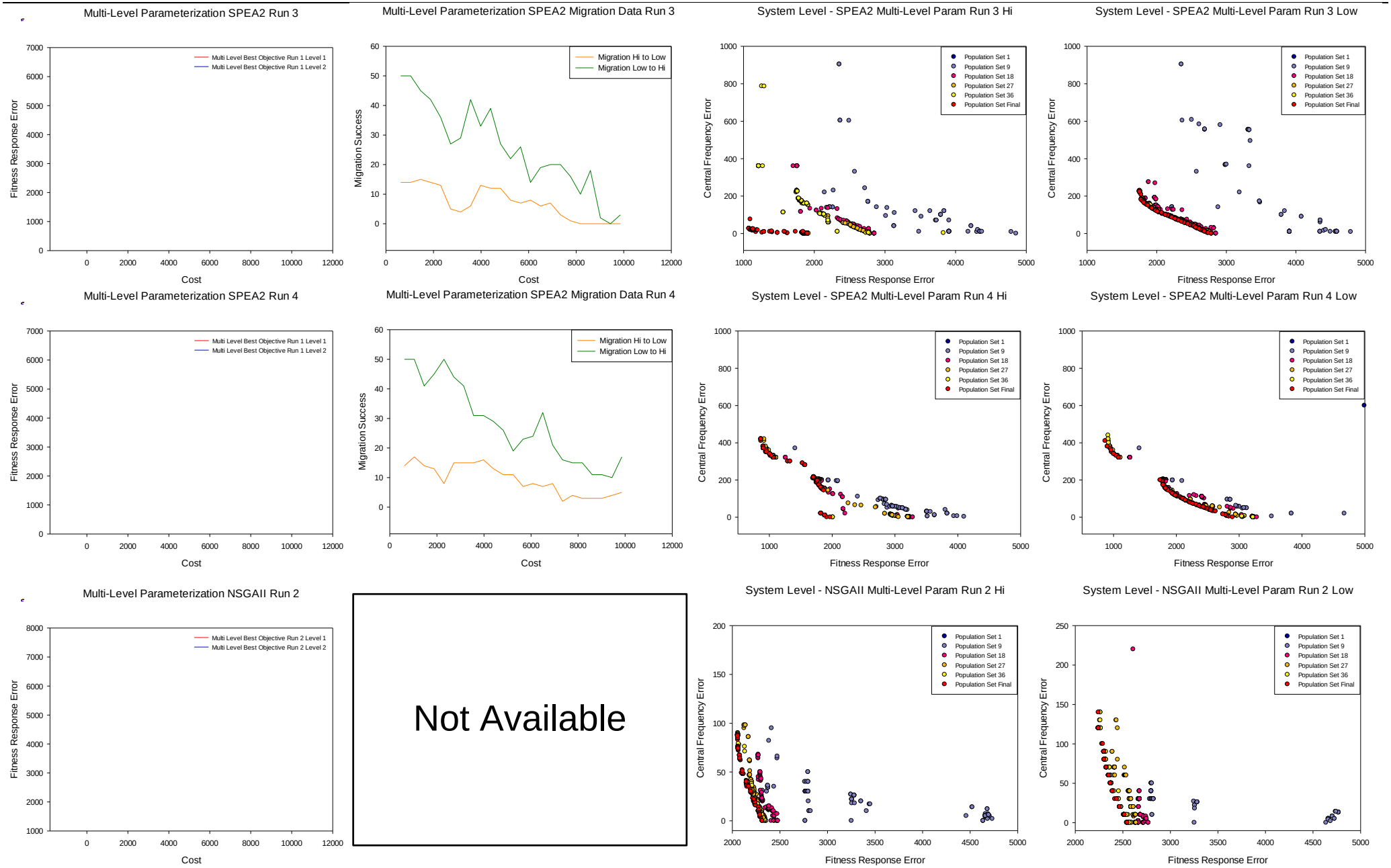


Figure 6.23 Multi-Level param experimental data includes: best filter response error objective, migration success data, population set generational data, for hi and low

6.3.6 Multidisciplinary optimisation

Large engineering design problems for example those found within the aeroplane industry can be difficult or impossible to undertake as a whole due to the large number of design variables, constraints and disciplinary analyses of the problem. In reality the design problem is often decomposed and each individual component solved or optimized separately by a design team, often focusing on specific variables, constraints and objectives.

MEMS are inherently multidisciplinary through the interaction of the mechanical and electronic components of the device. The application of MEMS into fields such as biology or chemistry through lab on-chip devices increases the number of disciplines a designer or design team must understand and integrate into the design process.

The principles of multidisciplinary design optimisation and in particular under an object based decomposition can be applied to both multiple and single levels of design in the MEMS hierarchical design process. The system level design of a bandpass filter consists of a single discipline in the form of electrical circuit simulation. The electrical equivalent circuit model contains equivalent elements for the mechanical resonator tanks and coupling springs that make up the bandpass device. Each of these components plays an important role in how the frequency transmission of the bandpass filter is shaped.

The decomposition of the system level bandpass problem into a number of subsystems each of which can be solved separately is the task of multidisciplinary optimisation strategy and it is outlined in figure 6.24. The system level design problem is decomposed into two subsystems, the first concerns the resonator tank components, and their effect on the central frequency and pass band characteristic of the frequency transmission. The second subsystem focuses upon the topology of the bandpass filter and its effect on the stopband / roll off and finally the coupling spring component and its effect on the bandwidth characteristic.

In order to optimise each of the subsystem problems, some new objectives are needed which focus on the new tasks outlined. The design of a filter which has a flat pass band characteristic within the target frequency range means the addition of a pass band error objective. This new objective is calculated exactly as in the filter response however, only the pass band is taken into consideration.

Subsystem two requires the construction of a bandwidth objective, the goal of which is to maximum the bandwidth of the first and last peaks of the filter transmission as shown in figure 6.25. The bandwidth is calculated as the distance in Hz between the first and last peaks of the bandpass filter divided by the average gain of the two peaks. Each peak is calculated simply as a point where either side shows a decline in the magnitude dB, and it must lie within the pass band range and have a magnitude between 15 dB and -15 dB. In the case where only one peak is present, then the bandwidth is set to a value of 1. The final objective for subsystem two is for stop band error and like the pass band error objective is calculated from the filter response.

In addition to the objectives a new constraint is added to the overall design process. Subsystem one contains a constraint to the total stop band error of the frequency transmission; this is to stop certain frequency transmissions from dominating at a detriment to the overall design

optimisation, these transmissions characteristically have a frequency response of 0 dB from start to finish of the bandpass target range giving them 0 pass band error, but large stop band error.

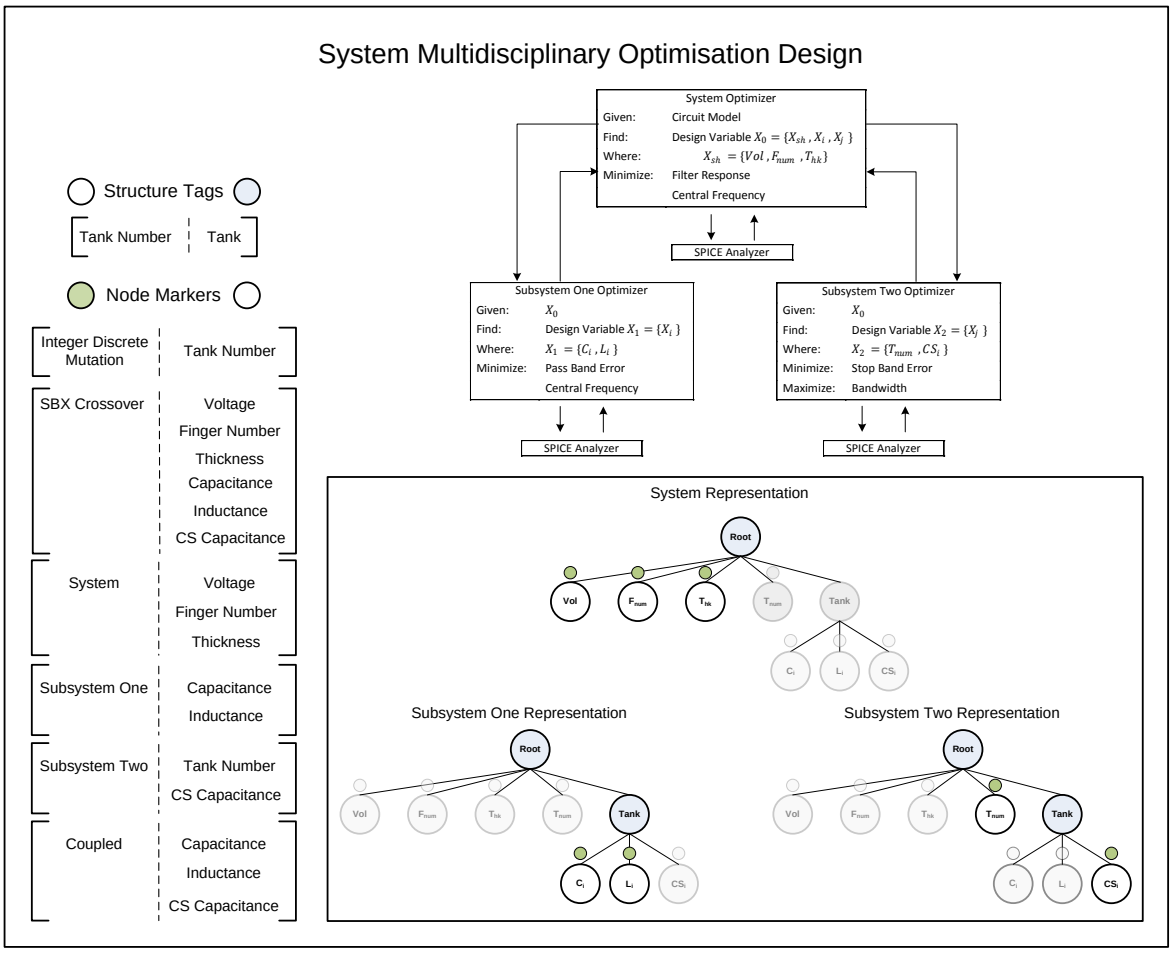


Figure 6.24 System multidisciplinary optimisation design template, with overview of problem, default representation, associated structure tags, node markers and global variables.

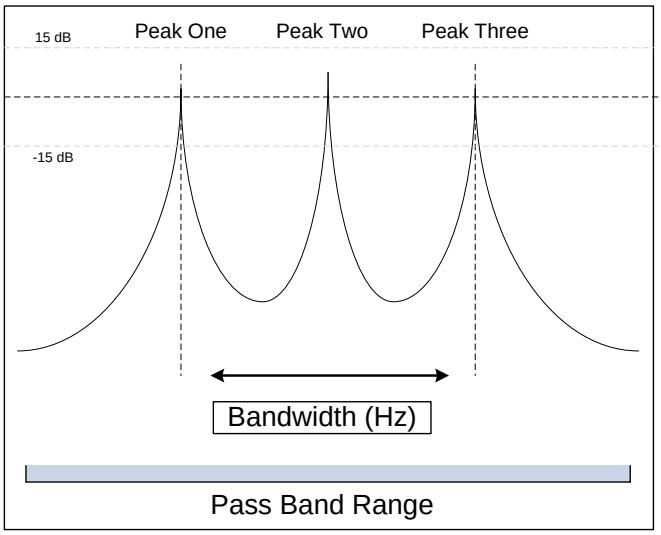


Figure 6.25 Bandwidth objective

Table 6.17 System Multidisciplinary Optimisation Objectives

	System Level		Subsystem 1		Subsystem 2	
	Objective 1	Objective 2	Objective 1	Objective 2	Objective 1	Objective 2
Objective Type	Minimize	Minimize	Minimize	Minimize	Minimize	Maximize
Objective Description	Filter Response	Central Frequency	Pass Band Error	Central Frequency	Stop Band Error	Bandwidth
Constraint Type	N/A		Inequality		N/A	
Constraint Description			Stop Band Error ≤ 1000			

The object based decomposition of the system level bandpass filter begins with classifying the customer requirements at the highest level, in this instance the characteristics of a bandpass filter with low insertion loss and high bandwidth over a target frequency range. The global objectives to try and meet these targets have already been outlined previously in the filter response and central frequency objectives. The decomposition of these objectives and the bandpass filter device into separate system and subsystem elements can then occur to try and aid the overall design process.

The definition of the functional behaviour of the bandpass filter and the components that make it up, and their role in the characteristics of the frequency transmission can now be undertaken.

The literature [131][274] points to the effect each individual resonator has on central frequency of the bandpass filter, both individual and coupled resonators and their mass, stiffness and damping values reflect their central frequency peak within the whole transmission shape. The pass band ripple and insertion loss are also heavily influenced by the constituent resonators that make up the bandpass filter [131]. Subsystem one is tasked with solving this particular functional requirement, with specific objectives and constraints as shown in table 6.17 and the decision variables for capacitance and inductance of each resonator tank which have a significant effect on these objectives being chosen.

The bandwidth of the filter device should be sufficient enough to cover the target pass band range, however the frequency transmission than needs to possess a sufficient roll off either side of the pass band into the stop band region to be effective. Both the bandwidth and stop band functional requirements are heavily influenced by the number of resonator tanks within the filter and the coupling spring stiffness that couple them [131][287]. Subsystem two contains objectives designed to focus on these particular functional requirements and this is matched with the chosen decision variables for coupling spring stiffness and tank number.

Often in complex multidisciplinary design problems such as this there are decision variables that can be equally shared among all lower subsystems. Here the system level contains three variables associated with the comb transducer and its effect on resistance values of each individual '*LCR*' tank. These are the voltage, thickness and finger number variables which are evolved at the system level and passed on through its population to all other subsystems.

The multidisciplinary optimisation strategy outlined in chapters 3 and 4 can involve the communication of coupled decision variables between one or more subsystems. Figure 6.24 shows the coupled decision variables within the multidisciplinary optimisation strategy, these being the capacitance, inductance and coupling spring variables of each resonator tank. Every

cycle after variation operators have acted upon the population of each subsystem, the gene swapper module transfers the coupled decision variables active in each subsystem representation to the other subsystems.

In essence genetic information is transferred and distributed between all the other subsystem offspring population sets. Genes or decision variable values which are functionally successful in all subsystems have a higher chance of surviving within the global 'genome' then those which survive in their local subsystem population set. The local objectives and constraints within each subsystem lead to a local selection pressure which each solution and its decision variables have to take into account.

The design process of the multidisciplinary strategy is essentially split between system and subsystem calls, and the cycle between them. Both the system and subsystem levels have the same default algorithmic parameters and population levels and therefore there is a choice into how many cycles each level is run in order to allow successful design optimisation within a budget of 10,000 functional evaluations. In the system level multidisciplinary optimisation design process the system level is run every 10 cycles, while each subsystem is run concurrently every cycle, this allows each subsystem to evolve its local population for 10 generations before the solutions are passed up to the system level.

6.3.7 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.26 and fully in appendix C.1. Also shown are the best frequency transmissions for each algorithm, ranked by the filter frequency objective in figure 6.27 and fully in appendix C.1 along with their objective values in table 6.15. Finally the hypervolume values for both algorithms are shown in table 6.18, with the best results shaded.

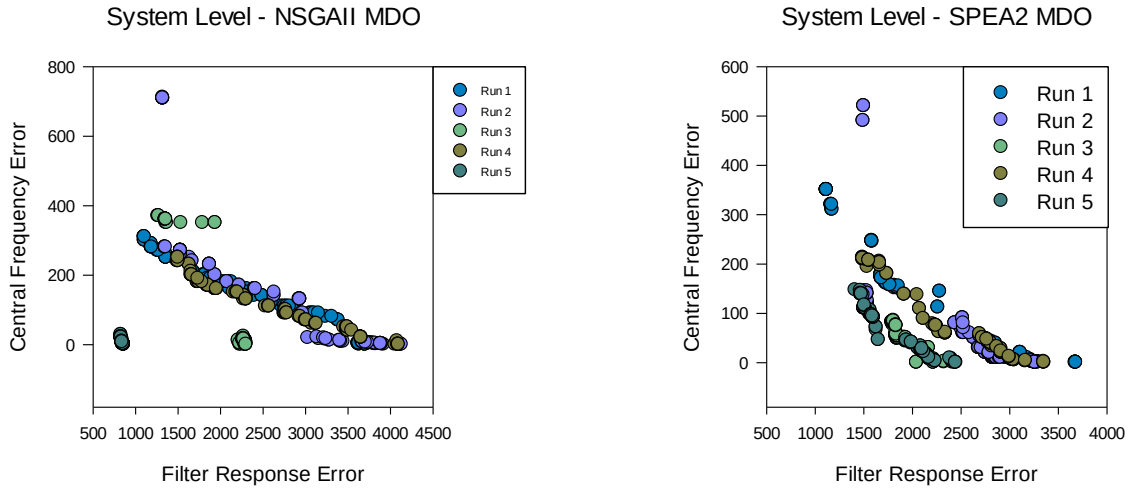


Figure 6.26 System level run 1 – 5 final population sets for (left) NSGAII multidisciplinary optimisation and (right) SPEA2 multidisciplinary optimisation

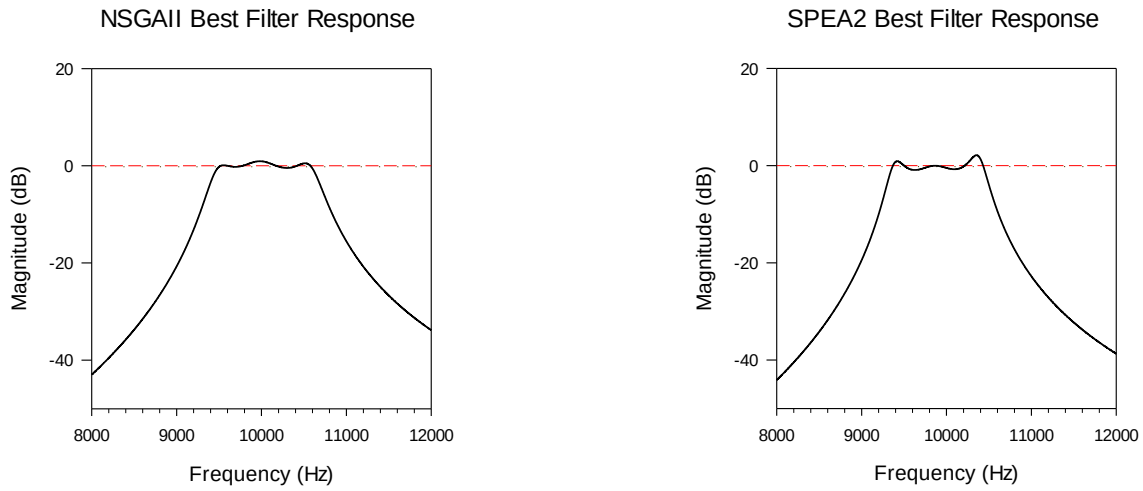


Figure 6.27 System level best filter response ranked by filter frequency objective for (left) NSGAII multidisciplinary optimisation and (right) SPEA2 multidisciplinary optimisation

Table 6.18 System Multidisciplinary Optimisation Bandpass Filter Results

System Multidisciplinary Optimization NSGAII					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	25	1103.233	310	183.27	3
2	28	1323.016	711	105.25	3
3	2	1268.895	370	133.04	3
4	9	1128.922	308	111.34	3
5	8	832.363	23	56.044	3
System Multidisciplinary Optimization SPEA2					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	4	1117.323	350	147.73	3
2	15	1493.894	490	177.23	3
3	4	1789.087	82	134.85	3
4	60	1489.723	212	13.81	3
5	15	1412.127	147	70.87	3

Table 6.19 System Level Hypervolume Results for NSGAII and SPEA2 MDO

System Multidisciplinary Optimisation		
Hypervolume	NSGAII	SPEA2
S^U	45837863.0641591	44084774.98745
S^M	44060899.260	42559736.279
S^L	43035839.2062174	41041427.413809

* (S^U S^M S^L) [10000, 5000]

Multidisciplinary optimisation of a MEMS bandpass filter through object decomposition formed the strategy to solve the system level MEMS design problem. Undertaken by both NSGAII and SPEA2 the characteristic performance of both has switched with the MOEA NSGAII outperforming SPEA2 in this example. Though the SPEA2 implementation performed similar to the single level strategy its results seem to indicate that this is possibly not due to convergence to a suboptimal front as was prevalent in the single level strategy (runs 2, 3 and 5). Instead the Pareto population sets for the SPEA2 MDO results are more fragmented, with both poor and

good fitness response 'front's in the objective space. The structured nature of the MDO strategy, with separate subsystems and objectives can result in an inefficient use of 'resources' or in this case functional evaluation cost as populations migrating from the system to subsystem levels need to be re-evaluated. This cost equates to about 1600 functional evaluations of lost search, a significant amount, which could have lead to the algorithm refining those transmissions from a 1400 filter response error to a lower one.

NSGAII on the other hand shows a robust performance over the 5 runs, significantly performing better than its single level counterpart as shown in table 6.18 and figure 6.26. An interesting characteristic of this approach is that a number of population fronts are continuous over a large range of the frequency fitness error objective. Whether the internal workings of the NSGAII algorithm and its crowding operator has had a positive effect on the solution spread which is filtered through to the MDO strategy in some way is a possibility though not certain and for brevity is left unexplored. The overall performance of both algorithms when compared against the single level strategies is shown in figure 6.28, with NSGAII outperforming the single level significantly and SPEA2 showing similar performance.

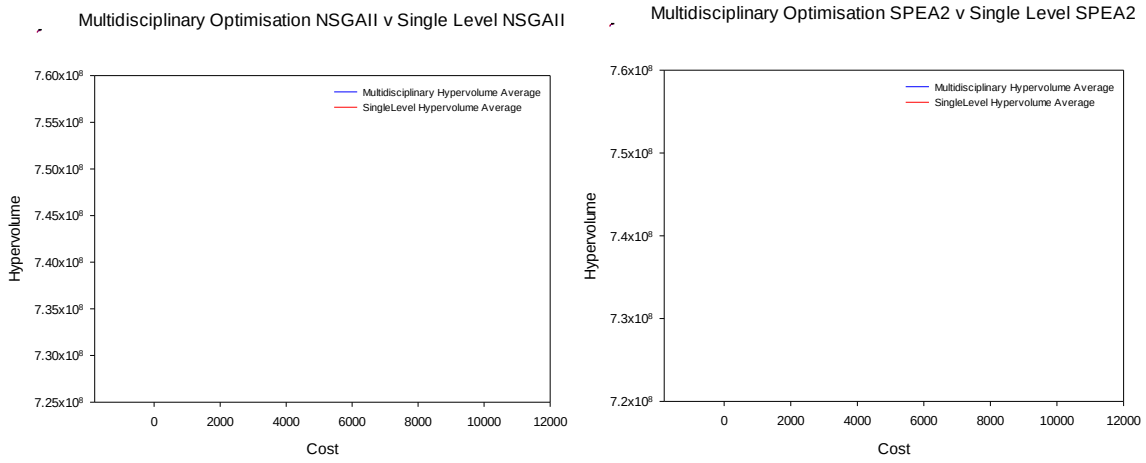


Figure 6.28 System level average hypervolume results for the 5 runs of the multidisciplinary optimisation and single level NSGAII and SPEA2 strategies * (S^U S^M S^L) [182000, 4150]

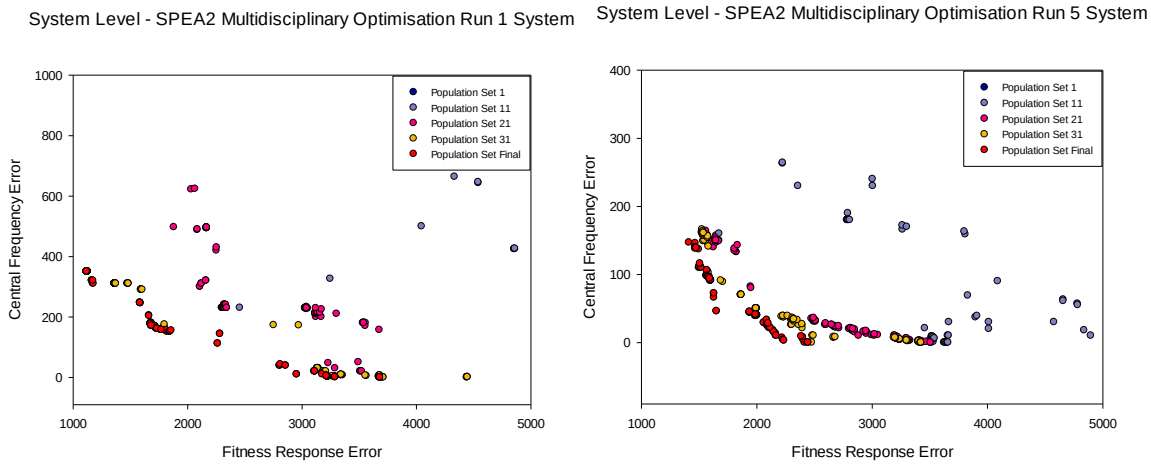


Figure 6.29 Generational system population plots for SPEA2 multidisciplinary optimisation runs 1 (left) and 5 (right) – each plot contains 5 equally distant generational plots.

Two examples of a generational system population set for SPEA2 MDO are shown in figure 6.29 showing each population set progressively moving towards the optimal in separated fronts indicating how each cycle contributes to Pareto objective fitness. There are little signs of convergence as seen in the single level SPEA2 examples of figure 6.5.

A major component of the MDO strategy is its structured system and subsystem hierarchy and the exchange of 'genetic 'material between each subsystem every generation. The partitioning of the design process into system and subsystem cycle events breaks up the design process as population sets are transferred from system to subsystem and vice versa. Figure 6.30 highlights the generational change to each subsystems population set through the individual hypervolume values of each run. The effect on subsystem one is negligible with steady improvement throughout the design process and typical of the system MDO level hypervolume results. However subsystem two shows a marked decrease in hypervolume or performance of the Pareto front after every system level 'update', where a new offspring population set is passed to each subsystem. This is in part due to the possible loss of good solutions through the passing of the offspring set, or the disparity of the genotype / phenotype of these solutions from the system level objective space, which uses the standard central frequency and filter response objectives, to subsystem two objectives of stop band and bandwidth. There is a slight exaggeration of the hypervolume 'dip' because solutions with a stop band of 0 are not found at the system level, these are evolved afterwards. The loss in the hypervolume performance of the population set is often linked to the bandwidth objective of subsystem two, where solutions at the system level often have smoother pass bands with peaks within the pass band range giving smaller bandwidths then those evolved locally before. The subsystem then has to search and evolve past solutions with an equivocal bandwidth.

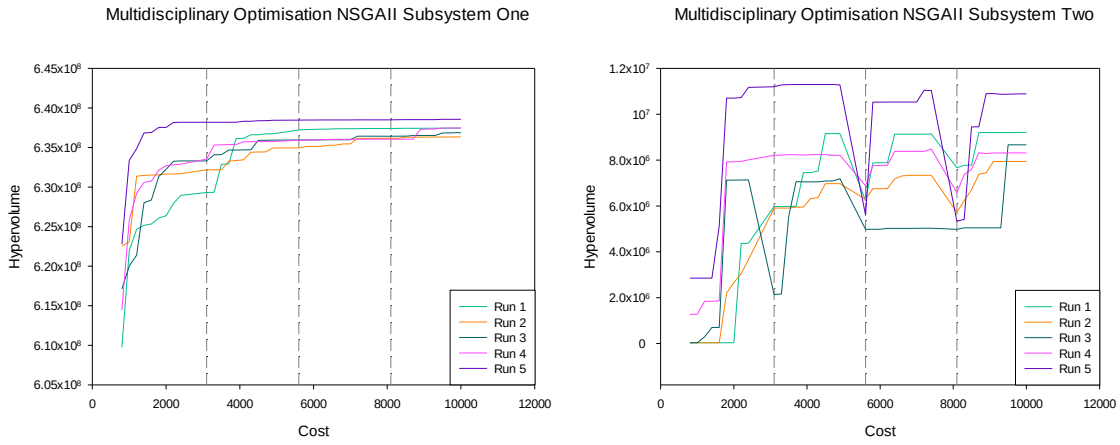


Figure 6.30 System level hypervolume results for the 5 runs of the NSGAII multidisciplinary optimisation strategy for subsystem one (Left) * ($S^U S^M S^L$) [4000, 160000] and subsystem two (right) * ($S^U S^M S^L$) [12000, 2.0].

Exploring the effect of each subsystem and the decision variables under their control and how the 'genes' and their 'alleles' evolve over the design process are presented next. Shown in figures 6.31 and 6.32 are a series of generational filter frequency transmissions for the best solution found in the system, subsystem one and two population sets for NSGAII MDO run 1 and SPEA2 run 1 respectively. Each solution chosen was ranked by the filter frequency error,

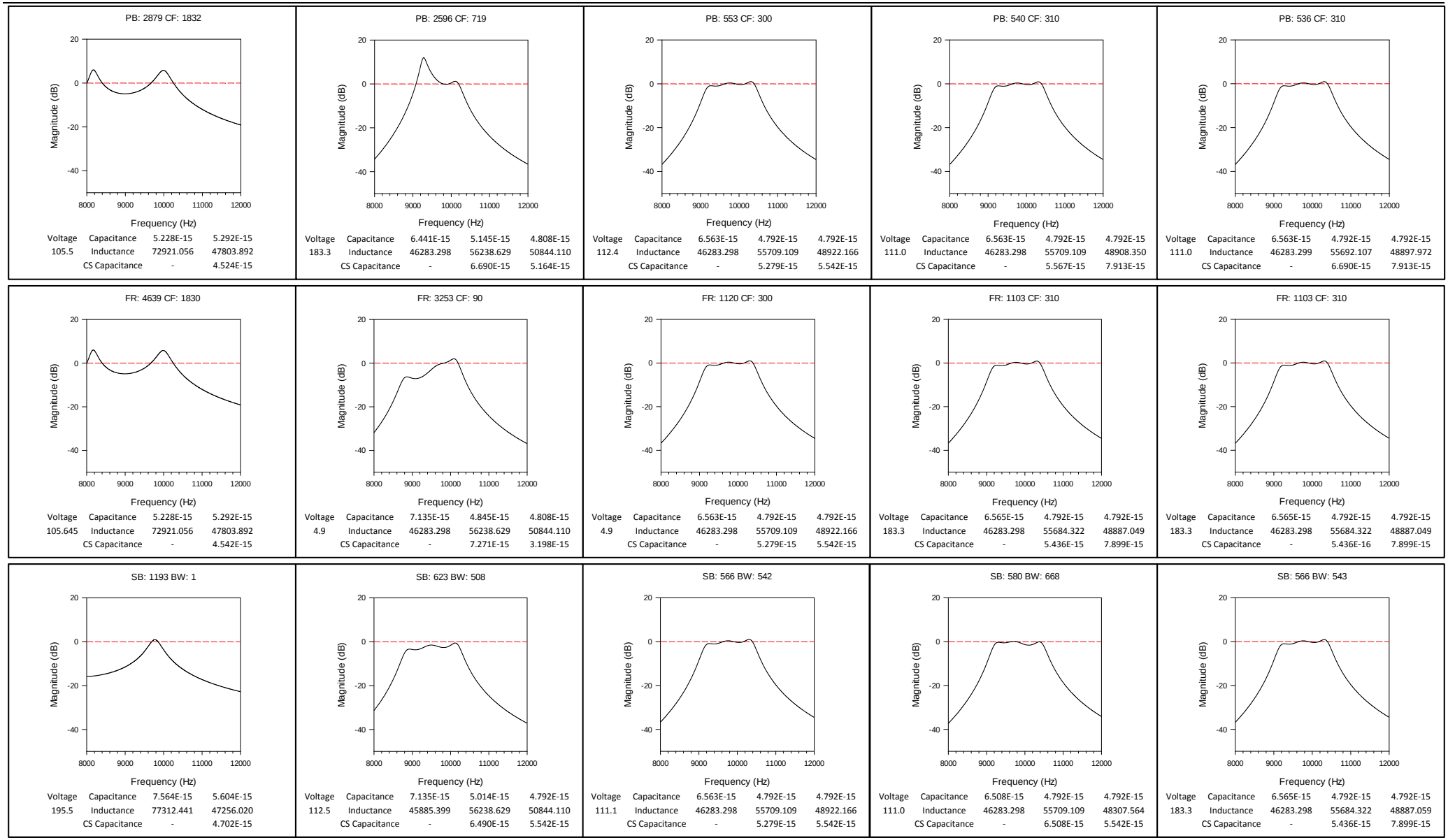


Figure 6.31 Best filter transmission plots for population (ranked by filter response objective), subsystem one (ranked by passband objective) and two best (ranked by bandwidth objective) over 5 generations (1, 11, 21, 31, 41) for NSGAII run 1. Each plot includes objective values and genotype values for each solution

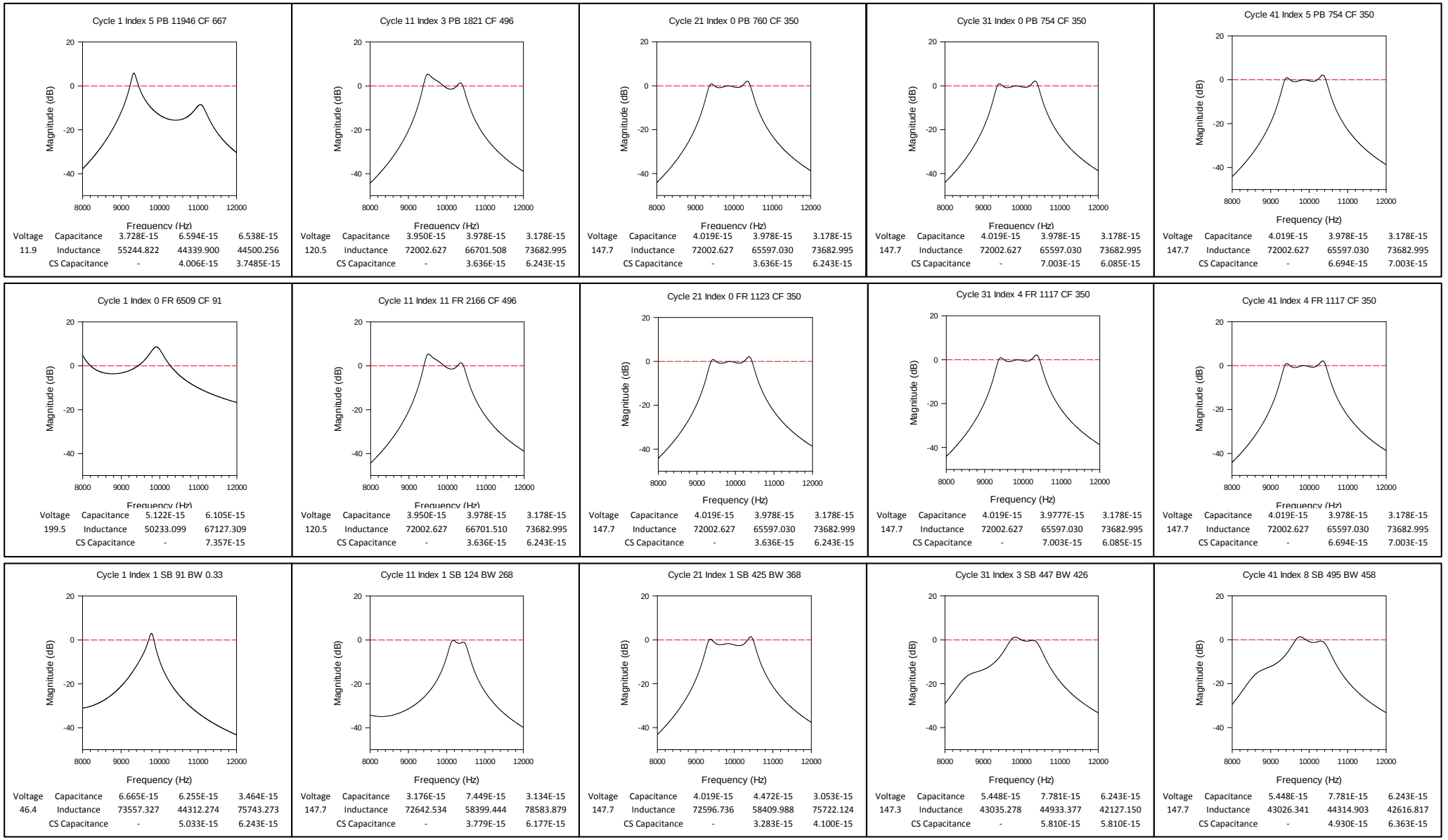


Figure 6.32 Best filter transmission plots for system population (ranked by filter response objective), subsystem one (ranked by passband objective) and two (ranked by bandwidth objective) over 5 generations (1, 11, 21, 31, 41) for SPEA2 run 1. Each plot includes objective values and genotype values for each solution.

passband error and bandwidth objective respectively and the objective values and tank values for each filter are shown below the response.

Linked to these filter plots are a series of density plots for each variable within the representation over the generations of the design process for NSGAII MDO run 1 and SPEA2 MDO run 1 held in appendix figures C.16 and C.17. Also included in figure C.16 are density plots for the local fitness values (pass band error, bandwidth) associated with each of the variables. These figures show the dynamic change in both the number of solutions in a population that contain a particular value for a variable over the course of the design process, but also whether that value leads to or is present in a high fitness individual.

The evolution of the frequency transmission, and its associated electrical equivalent circuit model values are impacted by the 'evaluation' criteria, in this instance the objectives, which are present in the different system and subsystem levels. From a design point of view, the objective was to allow subsystem one to focus upon the pass band and central frequency characteristics of the filter, and subsystem two to focus upon the bandwidth and stop band characteristics. As seen in the hypervolume figures 6.28 and 6.30 the majority of exploration and fitness improvement is undertaken early on in the design process and then followed with a convergence and plateau where little improvement is seen. The frequency transmissions for both examples NSGAII run 1 and SPEA2 run 1 show this with incremental improvement over the three separate partitions, cycle 1, 11 and 21, and followed with phenotypes that show minimal change over the following examples in cycles 31 and 41. The only deviation from this is in figure 6.32 and subsystem two results which switch to a phenotype which is locally better than that of the best global system level solution.

Looking at the specific subsystems, each one begins with the evolution of tailored frequency transmission characteristics, with subsystem one containing solutions focusing on the pass band region predominately with little regard for the stop band if only to remain unconstrained. The overall characteristic of the subsystem one solution in figures 6.31 and 6.32 takes on the shape of a typical bandpass filter, though with an unrefined pass band, at cycle 11 until convergence to the final phenotype at cycle 21.

Subsystem two focuses upon both bandwidth and the stopband region of the frequency transmission. The bandwidth of the frequency transmission between two or more peaks is established around cycle 11 in both NSGAII and SPEA2 examples and this is evolved to give wider bandwidths further on in the design process.

The next question is to what effect each subsystem and the solutions they evolve have on the global system level where the designer wishes to evolve the solutions they want to match the target filter characteristics. Interestingly in both examples the phenotypes of one subsystem match more closely the phenotype of the system level solution, here NSGAII subsystem two and SPEA2 subsystem one show closer affinity. The genotypic values in figures 6.31 and 6.32 of each of these solutions also show a close correlation, with individual capacitance and inductance values for each resonator tank closely resembling their system level counterparts, in particular SPEA2 subsystem one being identical.

The frequency transmission of the system level solution begins to match more closely with the best subsystem one solution from cycle 21 onwards, probably in part due to the pass band objective playing a more dominant part in the system level frequency error objective at this stage of the design process. However subsystem two frequency transmission results on a number of examples do not mirror the example found or retained at the system level.

The bandwidth of the subsystem two best solutions are often associated with the 2nd and 3rd peaks with the 1st outside the target passband range and therefore ignored. The effect of local selection pressure as a result of the subsystem objectives and any linkage between variables can play a role in the 'alleles' or values chosen in a particular solution.

The generational change in the density of each variable for the first 3 tanks and voltage variables are shown in appendix figures C.16 and C.17 for both NSGAII MDO and SPEA2 MDO run 1 experiments. In both experiments the density for the majority of variables shown are tightly grouped with little variation or migration along the variable bounds, this is shown by the long streaks over a number of generations. There is also a clear impact between the transition from system level to subsystem level population sets at generations 11, 21, and 31. Here for example in NSGAII run 1 subsystem one variables 1, 9, 11, 12 and 13 along with subsystem two variables 1, 6, 8, 9, 10, 12, 13 show the density for particular values are either introduced or 'bred' out as a result of the system level population.

The correlation between subsystem one and two variable densities are not always equal, the local selection pressure of each subsystem and perhaps other factors lead to different values or 'alleles' being dominant in each subsystem. The fitness or payoff of the variable values chosen can vary as well between subsystems with certain alleles having higher local payoff then in the other subsystem or showing equal payoff between the two objectives of pass band error and bandwidth.

6.3.8 System level comparison and analysis

The integration of MEMS into more complex commercial devices is only going to grow in the coming decades as new fields such as biology and chemistry are opened up with lab-on-chip devices. The function and utility of MEMS is also only going to increase further, often resulting in devices which contain many components and covering a number of multidisciplinary behaviour. The automated design synthesis and optimisation of these new MEMS devices will require the ability to model them in a number of levels of cost and accuracy so as to improve the practicality of automated design.

The system level of modelling and analysis presented to the designer contains tools for building more abstract representations of a MEMS device or component in order to allow functional analysis at a low cost in both time and resources though with a trade off in accuracy when compared with higher levels of design modelling [91]. A number of tools are present at the system level, be they block diagrams, bond graph representations or electrical equivalent models analysed through circuit simulators such as Spice [55]. A system level design problem was constructed based upon the synthesis and optimisation of a bandpass filter device modelled as an electrical equivalent model. A number of single and multi-level design optimisation

strategies were constructed and applied to this design problem in order to evaluate their efficacy towards optimisation and synthesis of a MEMS device.

The application of a multi-objective genetic algorithm towards the synthesis and optimisation of a bandpass filter has already been established in chapter 5 over a wide range of frequencies. A new bandpass frequency range was introduced and a single level strategy for both NSGAII and SPEA2 algorithms undertaken and compared against a number of multilevel strategies also using the NSGAII and SPEA2 algorithms. Table 6.20 holds the hypervolume performance of all the strategies employed for the system level design problem for both NSGAII and SPEA2 algorithms with the best and second best results shaded (dark/light) for each.

Table 6.20 System Level Hypervolume Results for Single and Multi-level Strategies for Both NSGAII and SPEA2 algorithms.

System Level NSGAII				
Hypervolume	Single Level	Multi-Level Evaluation	Multi-Level Parameterization	Multidisciplinary Optimisation
S^U	43755994.2233912	46307464.8585	45579384.4953333	45837863.0641591
S^M	39634066.167	44545442.897	40913176.976	44060899.260
S^L	32507262.7638433	40506807.6571	38342249.202637	43035839.2062174
System Level SPEA2				
Hypervolume	Single Level	Multi-Level Evaluation	Multi-Level Parameterization	Multidisciplinary Optimisation
S^U	44945558.0890751	45592440.8746	45595351.047224	44084774.98745
S^M	42433896.104	45033152.979	43962322.184	42559736.279
S^L	40276362.6859817	44162948.0661	41513986.5053333	41041427.413809

* (S^U S^M S^L) [10000, 5000]

It is clear from the results that the multi-level strategies outperform the single level strategy in both mean and boundary hypervolume values, in most cases greatly. The multi-level evaluation strategy produced the highest performance in terms of hypervolume and phenotype results with the ability to evolve filter shapes which matched the design targets set out over a large number of experimental runs. The second best strategy is mixed with multidisciplinary optimisation for NSGAII and multi-level parameterization for SPEA2 performing the best.

An important distinction between the two different levels of strategies is the use of a single or multiple set of populations / demes. The addition of multiple population sets can have distinct advantages, seen in a number of other research areas, for example ‘island genetic algorithms’ and ‘co-evolutionary Learning’. One clear benefit is the increase in ‘genetic diversity’ brought about through the relative isolation of each deme, though heavily influenced by the migration strategies employed. This diversity can alleviate any premature convergence of the total population allowing for further exploration of the design search space.

How much this and any other ancillary benefits such a multi population setup has to the overall performance increase of the multi-level strategies employed is not clear. There is the possibility to repeat the same multi-level strategies but using a single deme setup which essentially calls the various levels at different periods within the design process. Therefore to investigate the effect multiple demes have on performance, a single population multi-level evaluation strategy was employed to solve the system level design problem.

Here the process remains generally the same with three separate levels of analysis at different levels of cost and accuracy however only one population set remains within the design problem

structure. The design process begins with modelling and analysis at level one for a period of generations before such analysis is replaced with level two and then later level three analysis. The number of functional evaluations remains the same as the standard multi-level evaluation approach and table 6.21 holds information on the generation each level of analysis is invoked / replaced. Overall each level of analysis utilizes around a third of the total functional cost of the design process.

Table 6.21 System Level Multi-Level Evaluation Single Deme Analysis Level Ranges

Analysis Level	Generation Range
Level One	1 – 66
Level Two	67 – 133
Level Three	133 - 164

The new single deme design strategy was undertaken for both NSGAII and SPEA2 algorithms and the best phenotypes and hypervolume results for each run is shown in tables 6.22 and 6.23 respectively. Figure 6.33 shows the generational performance of both algorithms in comparison with the standard multi-level evaluation methodology. The hypervolume values require some scaling for values from level one as the objective values for the filter objective error contain only a third of the value of the higher level analysis due to the lower number of data points sampled. Therefore all filter error objective values at this level were multiplied by three in order to give some parity to the default level three analyses. The transition from each level shows little to no change in hypervolume characteristics so the scaling should somewhat accurately reflect the true value throughout the whole design process.

Table 6.22 System Multi-Level Evaluation Single Population Bandpass Filter Results

System Multi-Level Evaluation Single Population NSGAII					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	98	884.223	350	49.54	3
2	8	2570.727	70	60.02	3
3	1	1415.210	206	146.63	3
4	0	780.022	559	145.38	3
5	2	851.282	610	11.16	3
System Multi-Level Evaluation Single Population SPEA2					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	71	1236.113	690	25.95	3
2	3	918.613	650	102.87	3
3	39	895.602	540	64.70	3
4	0	808.453	460	21.38	3
5	1	915.818	155	159.48	3

Comparing the hypervolume results with the standard multi-level evaluation strategy in table 6.12 shows a drop in performance for the NSGAII algorithm in part due to the poor performance of runs two and three. The SPEA2 algorithm is similar in performance with a slight improvement in the average hypervolume values over the five runs. The generational hypervolume plots show agreement with this assessment and overall there is now real net gain in performance over the

two algorithms for the multi-level evaluation strategy using a single or multi population methodology for this small set of experiments.

Table 6.23 System Level Hypervolume Results for NSGAII and SPEA2 Multi-Level Evaluation Single Population Strategy

System Multi-Level Evaluation Single Population		
Hypervolume	NSGAII	SPEA2
S^U	45703085.7470257	45939251.6608935
S^M	43321409.05290	45035741.61141
S^L	37140511.4763832	43552577.1334492

* (S^U S^M S^L) [10000, 5000]

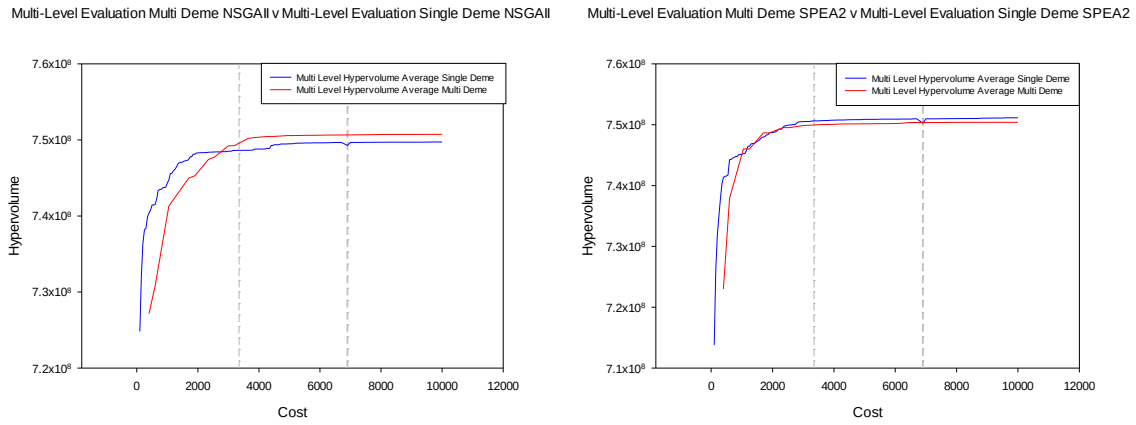


Figure 6.33 System level average hypervolume results for the 5 runs of the multi-level Evaluation multi deme and single deme NSGAII and SPEA2 strategies * (S^U S^M S^L) [182000, 4150]

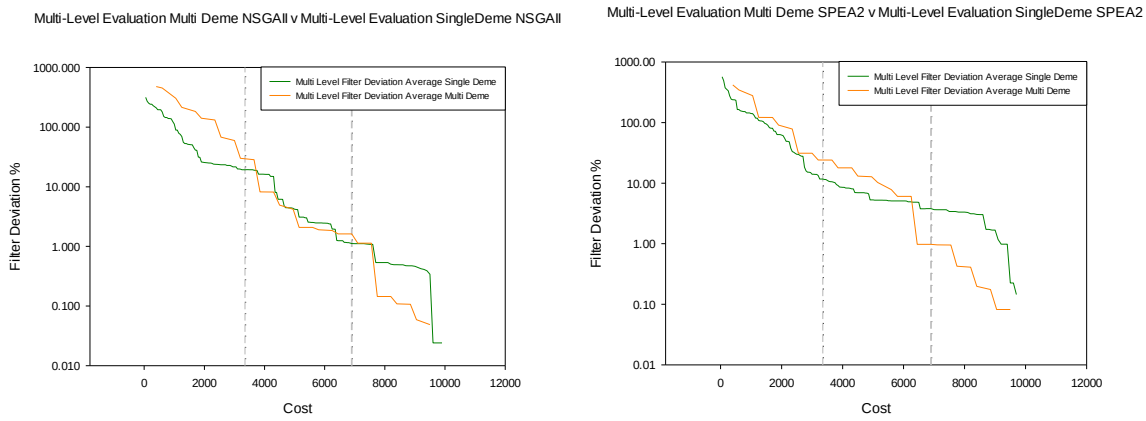


Figure 6.34 System level average filter deviation % from best solution results for the 5 runs of the multi-level evaluation multi deme and single deme NSGAII and SPEA2 strategies

Though the overall performance of the single deme method is not too dissimilar from the multi deme method there is a characteristic change to the early performance in the design process. This is perhaps unsurprising given the additional functional evaluations the reduced computational cost level one brings. However there is also the possible benefit brought about through changes to the objective space as a result of the more granular analysis and the effect this has on how the optimizer is able to vary offspring. One of the drawbacks of the default model found more so in the single level strategies is the inability to break away from certain

phenotypes and essentially become stuck in a certain region of the objective space. The more granular analysis could possibly benefit variation by allowing larger genotypic changes to occur without as drastic a change in phenotype characteristics and objective space values allowing solutions to transverse the search space more easily than would otherwise occur using the default, accurate model.

Figure 6.34 shows the average deviation percentage from the best filter error objective found in each run over the course of the design process. What is clear in both this figure and figure 6.33 is that a large proportion of the hypervolume performance is evolved early on in the design process, and in the case of the deviation from the best solution found, solutions within 10% are evolved at the level one design process region. Looking at both the NSGAII and SPEA2 algorithms it would seem that the single deme approach which exclusively uses the lowest level of analysis earlier on also improves filter error objective values faster than the multi deme approach. It also highlights a possibly avenue for increased performance with a faster improvement to the filter shape early on it than may prove beneficial to switch to a separate local gradient–based optimizer to improve what is the remaining 10% optimization faster than would otherwise be possible with a more global stochastic method used here.

The application of both multi-level and multidisciplinary optimisation strategies to the optimisation of a system level MEMS design problem has proven to be successful. The next section looks to apply the same strategies to a MEMS design problem set within the device level of modelling and analysis.

6.4 Device level design optimisation

The bulk of automated MEMS design optimisation within the literature occurs at the device level, often through 2D analytical or NODAL modelling and analysis. A number of examples of device level optimisation have already been undertaken in chapter 2 and one of these in particular, the folded flexure resonator, forms the basis for the next level of design optimisation.

A major component of the bandpass filter, the folded flexure resonator, consists of functional values for example mass and stiffness which are important design targets in the synthesis of a bandpass filter. Chapter 4 outlined an example of resonator design optimisation for the target values of mass and stiffness in relation to the electrical equivalent values obtained at the system level of a bandpass filter. This is extended now into a three objective design problem, similar to the previous chapter but with an additional cost objective, in this case through the minimization of the total area of the device. Often designers have to perform what is more accurately synthesis, i.e evolving a default device model to match certain targets, along with traditional optimisation, in this case the total area of the device, an important characteristic in reducing cost. The design synthesis objectives though one of many, are often the most important, improved performance of cost objectives mean very little if the device does not match required synthesis goals, as is the case with mass and stiffness. It is a possibility to replace such synthesis targets with constraints, but this would remove any Pareto choice for the designer that would emerge from these two objectives.

The modelling and analysis of the folded flexure at a default single level analysis involves the use of NODAL modelling and the simulator Sugar [66] as utilized in chapter 4. However an important

change has been made in anticipation of the multi-level evaluation strategy. The evaluation of the device using two separate and different approaches has already been undertaken, with the lumped parameter analytical modelling found in chapter 2 and the NODAL modelling in chapter 4. However the physical 2D layout and their associated design variables are different in both cases, with the analytical model having a simplified central mass structure in comparison to the NODAL model. To allow easier transition of the representation from one level to another the NODAL model has been simplified to match the analytical model more closely as seen in figure 6.35.

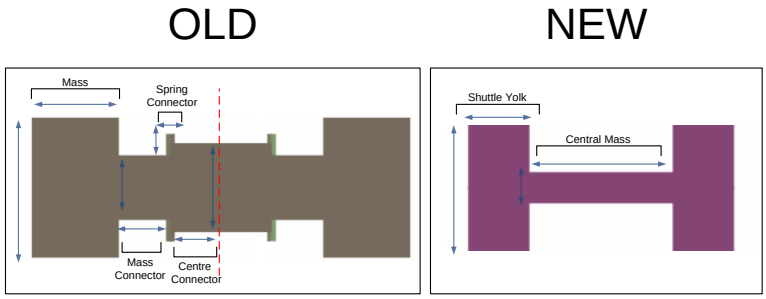


Figure 6.35 Folded flexure resonator central mass layout changes from previous design to current one.

Another addition to the folded flexure resonator problem is the changes to the springs that make up the folded flexure component. Rather than consisting of a single beam replicated for all 8 springs, the representation is expanded to allow up to 6 individual beams to be sequentially connected for each spring. The folded flexure component is then mirrored in the x and y axis to fill the remaining positions. This complexity should hopefully open up more of the design search space and allow for more complex or novel topologies and shapes to arise to meet the design targets. This change is filtered to the structural crossover module employed in the previous folded flexure design problem and now updated to handle inter beam crossover between selected beams as shown in figure 6.36.

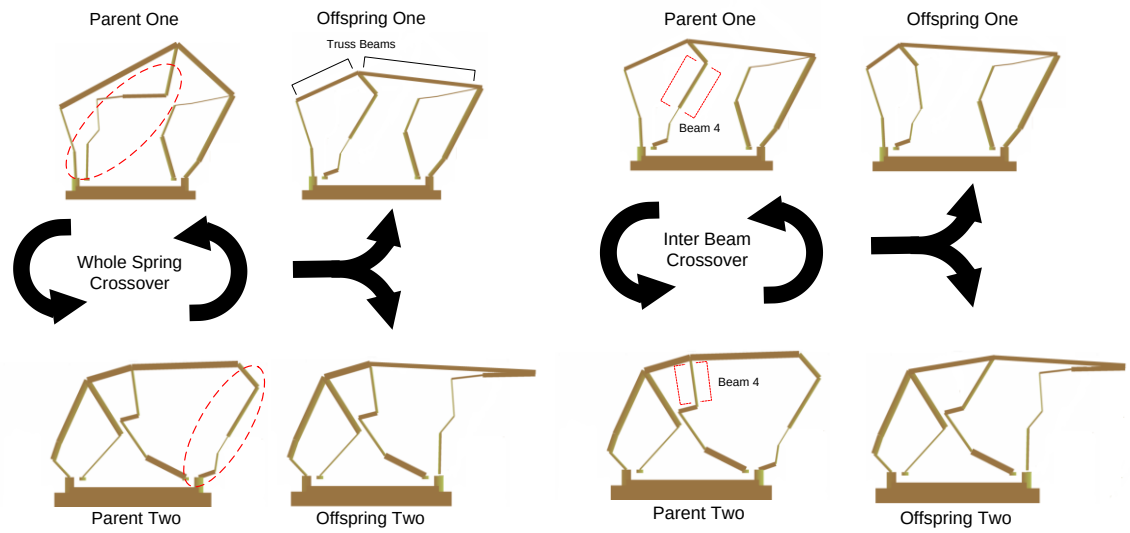


Figure 6.36 Whole spring (left) and Inter beam (right) crossover between two selected solutions

The device level decision variables are shown in table 6.24 and include the changes to the central mass structure and folded flexure component with new variables for each. Also included is a variable for the thickness of the device to better aid reduction in total area when also looking to evolve designs which match a target mass value. New objectives and constraints are also added, the first of which simply deals with the central mass and insures the shuttle yolk is greater in length then the central mass section. The addition of an intersection constraint is a result of the new fidelity of the spring structures and the interconnected beam segments that can result within them. The intersection constraint is designed to check for beam crossover between two beams, generally between two springs as shown in figure 6.37. In order to test for intersection, the principles of the separating axis theorem [291] between two polytopes, in this case rectangular bounding boxes is used. A special case is included for two neighbouring beams on the same spring; this is to allow such beams to be connected without failing the intersection test by reducing the bounding box size of the beams by 15%. This still allows for certain characteristics such as a small angle between the two beams and the resulting overlap to be considered constrained, such sharp angles can lead to stress concentrations in real devices and the possibility that the shape itself does not mirror real life as the NODAL analysis does not take into account such overlap.

Table 6.24 Device Level Problem Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Central Mass Length (μm)	Real Valued	10	600
Central Mass Width (μm)	Real Valued	10	400
Shuttle Yolk Length (μm)	Real Valued	2	400
Shuttle Yolk Width (μm)	Real Valued	10	400
Beam Number	Integer	1	6
Beam Angle	Real Valued	45	135
Beam Length (μm)	Real Valued	5	100
Beam Width (μm)	Real Valued	2	20
Truss Width (μm)	Real Valued	2	20
Anchor Placement Length (μm)	Real Valued	0	400
Thickness (μm)	Real Valued	2	200
Objectives		Constraints	
Folded Flexure Stiffness Kx Error	Minimize	Constraint 1	Inequality > 0
Folded Flexure Mass Error	Minimize	[Shuttle Yolk Length - Central Mass Width]	
Total Area	Minimize	Intersection Check	Inequality

Table 6.25 Device Level Synthesis Objective Target Information

Synthesis Objective	Target Value
Folded Flexure Stiffness Kx Error	2.45 N/m
Folded Flexure Mass Error	5.12E-10

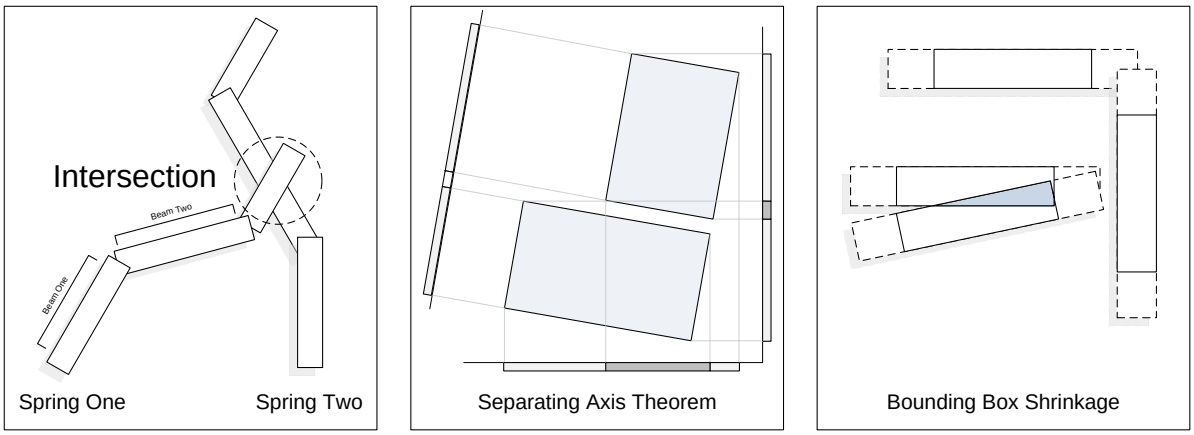


Figure 6.37 Intersection, separating axis theorem and bounding box shrinkage alterations

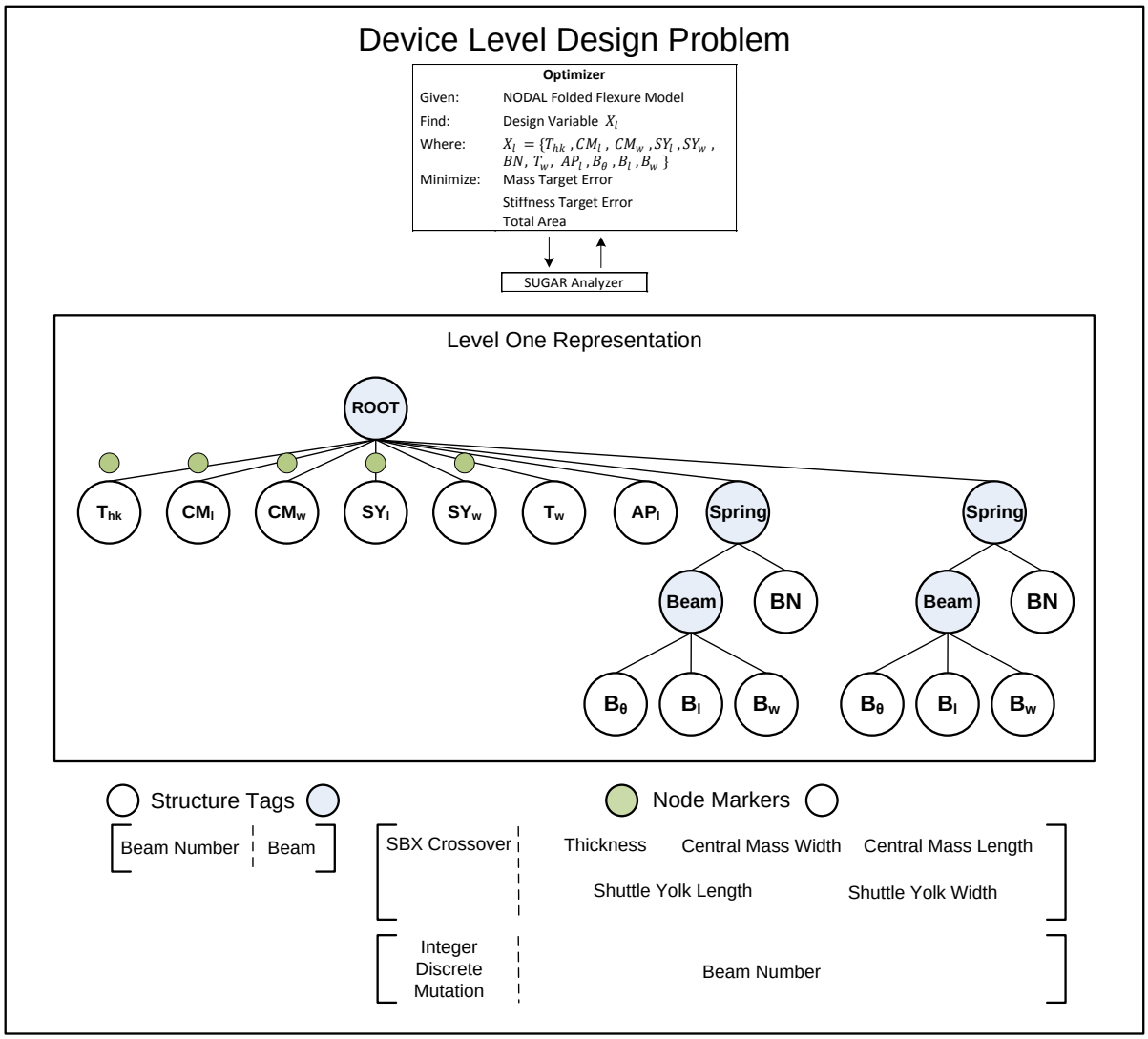


Figure 6.38 Device level design template, with overview of problem, default representation, associated structure tags and node markers.

The overall device level design problem is shown in figure 6.38 along with the structure and node markers associated with the updated representation. The updated representation variable bounds and design problem objectives and constraints are shown in table 6.24. Algorithmic parameters for both NSGAII and SPEA2 remain unchanged from those used in the system level design, and the total functional evaluations remains 10,000. The targets for both the stiffness and mass objectives are shown in table 6.25 and come from a previous design found at the system level and converted to its mechanical equivalent values.

6.4.1 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.39 and fully in appendix C.2. A list of the best results ranked by total area and constrained to having a stiffness and mass error of less than 1% from the target for each run are shown in table 6.26 and their phenotypes shown in figure 6.40. Finally the hypervolume values for both algorithms are shown in table 6.27, with the best results shaded.

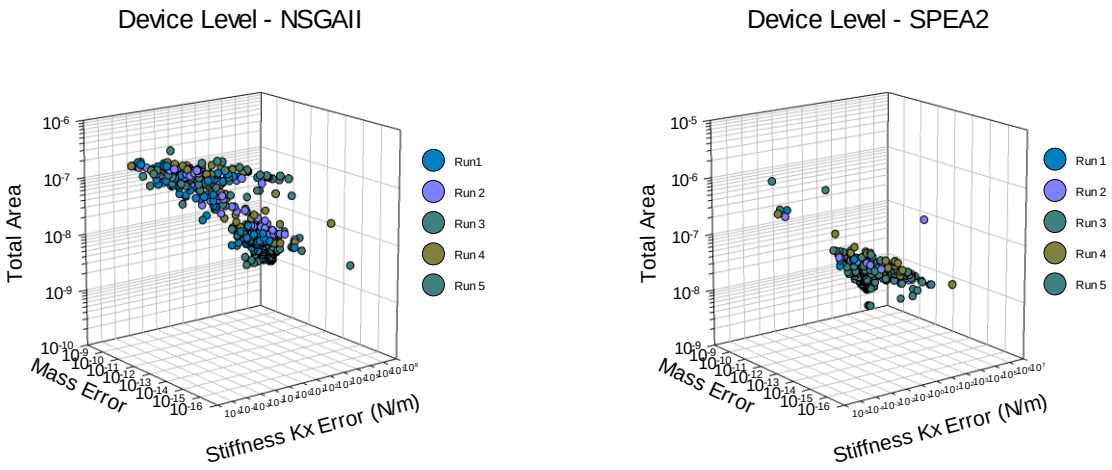


Figure 6.39 Device level run 1 – 5 final population sets for (left) NSGAII and (right) SPEA2

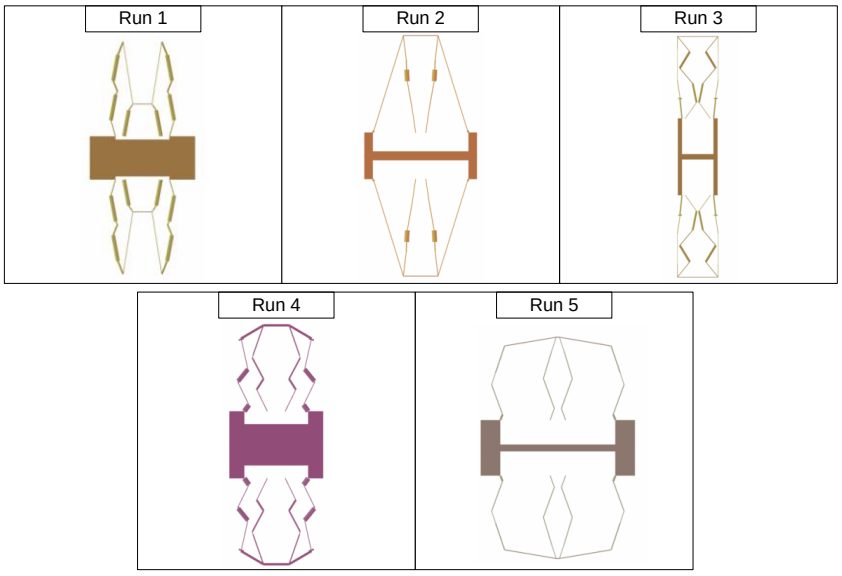


Figure 6.40 Device level run 1 – 5 folded flexure best result ranked by total area for NSGAII

Table 6.26 Device Level best results ranked by total area for solutions with ≤ 1% error

Device Level NSGAII					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	Mass Error	Total Area
1	100	1	0.002297499	2.9771955E-14	2.339771E-07
2	100	4	0.016984615	5.6951316E-13	2.195535E-07
3	100	1	0.01378337	9.6188685E-14	1.99696931E-07
4	100	1	0.02305925	2.5184002E-13	2.20929428E-07
5	100	2	0.022463708	4.6408605E-13	3.21718333E-07
Device Level SPEA2					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	Mass Error	Total Area
1	100	0	Not Applicable		
2	100	0			
3	100	0			
4	100	0			
5	100	0			

Table 6.27 Device Level Hypervolume Results for NSGAII and SPEA2

Device Level		
Hypervolume	NSGAII	SPEA2
S^U	1.29963645850762e-007	1.29929836220047e-007
S^M	1.2994629E-07	1.2985024E-07
S^L	1.29929021045027e-007	1.2974065198587e-007

* (S^U S^M S^L) [$1.3e^7$, $1e^{-9}$, $1e^{-5}$]

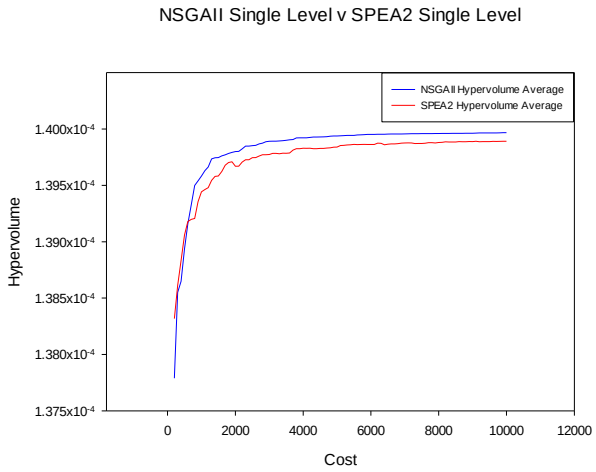


Figure 6.41 Device level average hypervolume results for the 5 runs of the single level NSGAII and SPEA2 strategies * (S^U S^M S^L) [1.4e7, 1e-6, 1e-5]

The single level design optimisation of a device level MEMS folded flexure resonator by both SPEA2 and NSGAII has been undertaken. There is a clear distinction on the performance between the NSGAII and SPEA2 MOEAs both in the overall Pareto sets and their associated hypervolumes which show overall superiority by NSGAII as seen in figure 6.41 and table 6.27. Particular importance is placed on the synthesis objectives of stiffness kx and mass error where a deviation of at most 1% from the target values is a design goal. NSGAII was able to find a solution which matched these targets in all 5 runs though at low numbers, while SPEA2 failed to find a single solution which could match these design goals. A possible explanation for these

differences lies in the regions of the objective space in which both algorithms explore and exploit. Both NSGAII and SPEA2 focus heavily on the total area objective particularly in the region of 10E-8 to 10E-9 total area. This can have a number of possible drawbacks on the other objectives, with a strong correlation between reducing total area and increasing the stiffness error values of solutions, and a similar effect on the mass error as seen in figure 6.39.

In a multi-objective problem each objective is often competing against the others giving rise to the Pareto set and the final decision as to which solution to choose at the end. There has to be some level of 'cooperation' between the objectives so neither one is too disruptive to the others as to be detrimental or fatal to finding solutions which show good performance across all objectives.

The design optimisation of a folded flexure resonator has been undertaken here and in chapter 4 as both a two and three objective problem, with both synthesis and cost objectives. Looking at the results of the two objective problem show the ability of the optimizer to solve for both stiffness kx and mass error objectives, and find solutions with an error less than 1% successfully and robustly over a number of different runs. There is evidence to show that antagonism between the two objectives is small. This is in contrast with the three objective problem, here total area as an objective seeks to alter the topology and sizing of the device in such a way as to increase the difficulty for the optimizer to focus on the other objectives. In figure 6.42 an example of a topology with a very small total area highlights the changes to the folded flexure structure necessary to achieve this and the increase in thickness and the rigid flexure give rise to a very large stiffness.

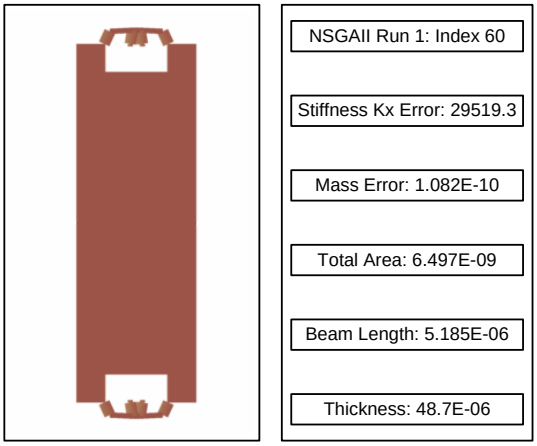


Figure 6.42 Folded flexure resonator example with reduced total area from NSGAII run 1

Design of the resonator structure involves two main components, the folded flexure spring and the central mass unit. Each of these has a profound effect on the objectives for stiffness and mass and therefore must cooperate through the design process so as not to disrupt either objective. Variations to the central mass can influence the folded flexure structure and ultimately the stiffness value of the resonator, mainly through central mass length. Equivocally the folded flexure spring structure itself contributes to the overall mass of the resonator. However each influence is not as profound on the others objectives, for example the central mass is on the mass objective, therefore the central mass variables have a larger control on the destiny of the mass objective, while the folded flexure variables control the stiffness objective.

The addition of the total area objective can be disruptive as both folded flexure and central mass variables have a direct influence on this particular objective. Simply reducing the variables to their lower bounds, in particular central mass length and beam length, can give rise to phenotypes which exhibit an extremely small total area. However this is bound to affect both of the other objectives, with the target mass there is a clear bound on how small the device can go before it deviates away from the target, while compact and rigid folded flexure springs often give rise to very high stiffness values, a deviation from the stiffness objective set out. The total area objective at first glance is the easiest to undertake, while both mass and stiffness objectives the hardest as they are synthesis objectives, targets for which the resonator must seek out and hover around. In particular the topology of the springs and their sizing values of their constituent beams have a large influence on stiffness.

Once a population has established a large number of solutions within the objective space of very low total area of which the individual topologies and sizing of the solutions are similar to those shown in figure 6.41 than it can be hard or impossible for the necessary functional change in the mass and flexure components to occur.

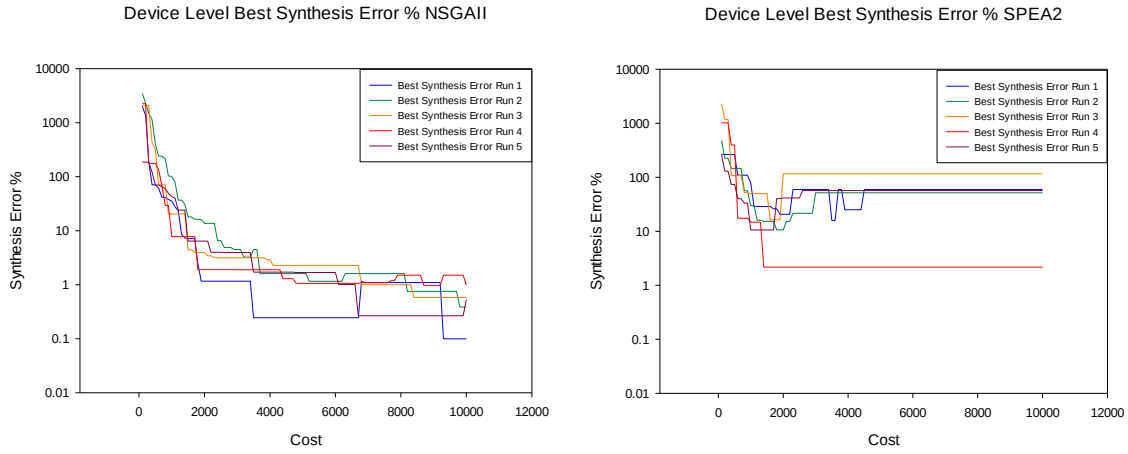


Figure 6.43 Best synthesis error percentage values for each experimental run of device level (left) NSGAII and (right) SPEA2

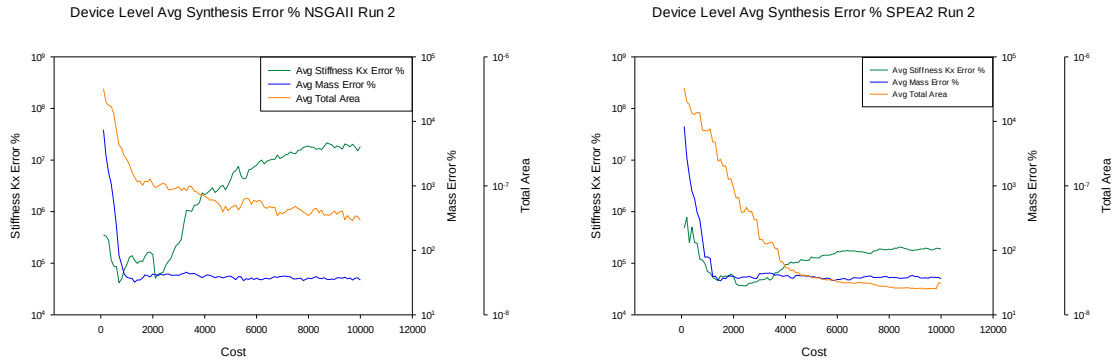


Figure 6.44 Average synthesis error for stiffness kx and mass objectives and average total area for (left) NSGAII and (right) SPEA2 run 2.

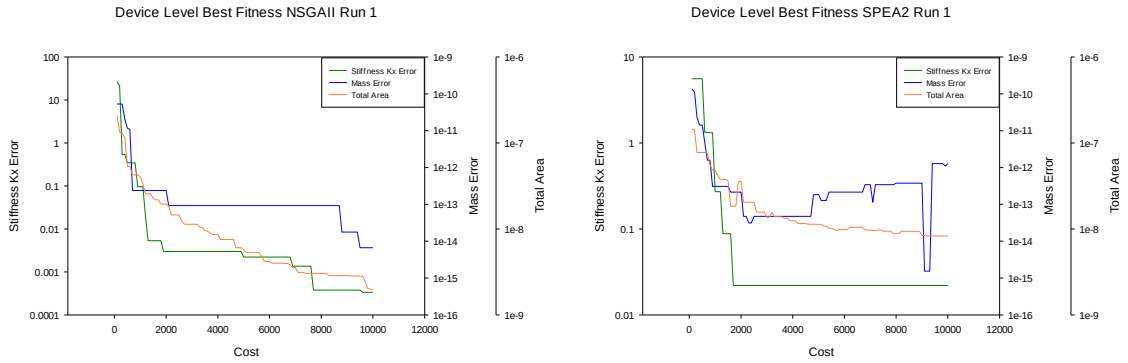


Figure 6.45 Best fitness objective for device level NSGAII (left) and SPEA2 (right) run 1

It is important for both algorithms to find solutions which contain an error of less than 1% for both the stiffness kx and mass objectives. Figure 6.43 shows the total error for the best solution of these two objectives for each run of the NSGAII and SPEA2 algorithms over the design process. Here NSGAII shows a progressive improvement over the generations with a slight degradation as the total error gets smaller near the end of the design process. SPEA2 however shows a completely different characteristic with some improvement up to and around a 10% total synthesis error after which there is a sudden reverse with the synthesis error increasing and then stalling at a particular value. Both algorithms exhibit loss of performance in the best solution, however NSGAII seems able to recover and continue improvement while this is not the case with SPEA2 which stalls. Factors within both algorithms must play a role in this behaviour, figure 6.44 shows the average synthesis error and total area objective values for the populations of run 1 for both NSGAII and SPEA2. Though only single examples, each of the 5 runs of both algorithms exhibit similar characteristics, in particular SPEA2 populations converge to a much smaller average total area than NSGAII, both show an improvement in the average stiffness kx error before increasing over the generations, with NSGAII expanding this error to a much larger degree, and both algorithms show similar synthesis error for the mass objective.

The percentage stiffness kx error for both algorithms is high as a result of the focus placed upon the total area objective. Looking at figure 6.43 it would seem that SPEA2 should have a higher average stiffness error given the average total area values for the population. However in the context of the population sets in figure 6.39 it can be seen that NSGAIIs ability to spread the population over the total area objective while SPEA2 focuses on a particular band can lead to a higher mean total area value for NSGAII while SPEA2 is more concentrated. In fact NSGAII is able to fill the total area objective space with solutions which have a much lower value than those of SPEA2. Figure 6.45 shows an example of the best objective values obtained over the generations for run 1 of both NSGAII and SPEA2. Individually each of the synthesis objectives are evolved to have very low error, however it is only in NSGAII where it is successfully found within a single individual at sufficiently low levels. A series of generational results for the best solution ranked by the total synthesis error % for NSGAII runs 1, 2 and 5 can be seen in figure 6.46.

Looking over the results and further analysis it can be seen that there is a particular focus on the total area objective, and that maintaining solutions with a relatively low synthesis objective error is low or nonexistent, and this is possibly as a result of the antagonism between the synthesis and cost objectives.

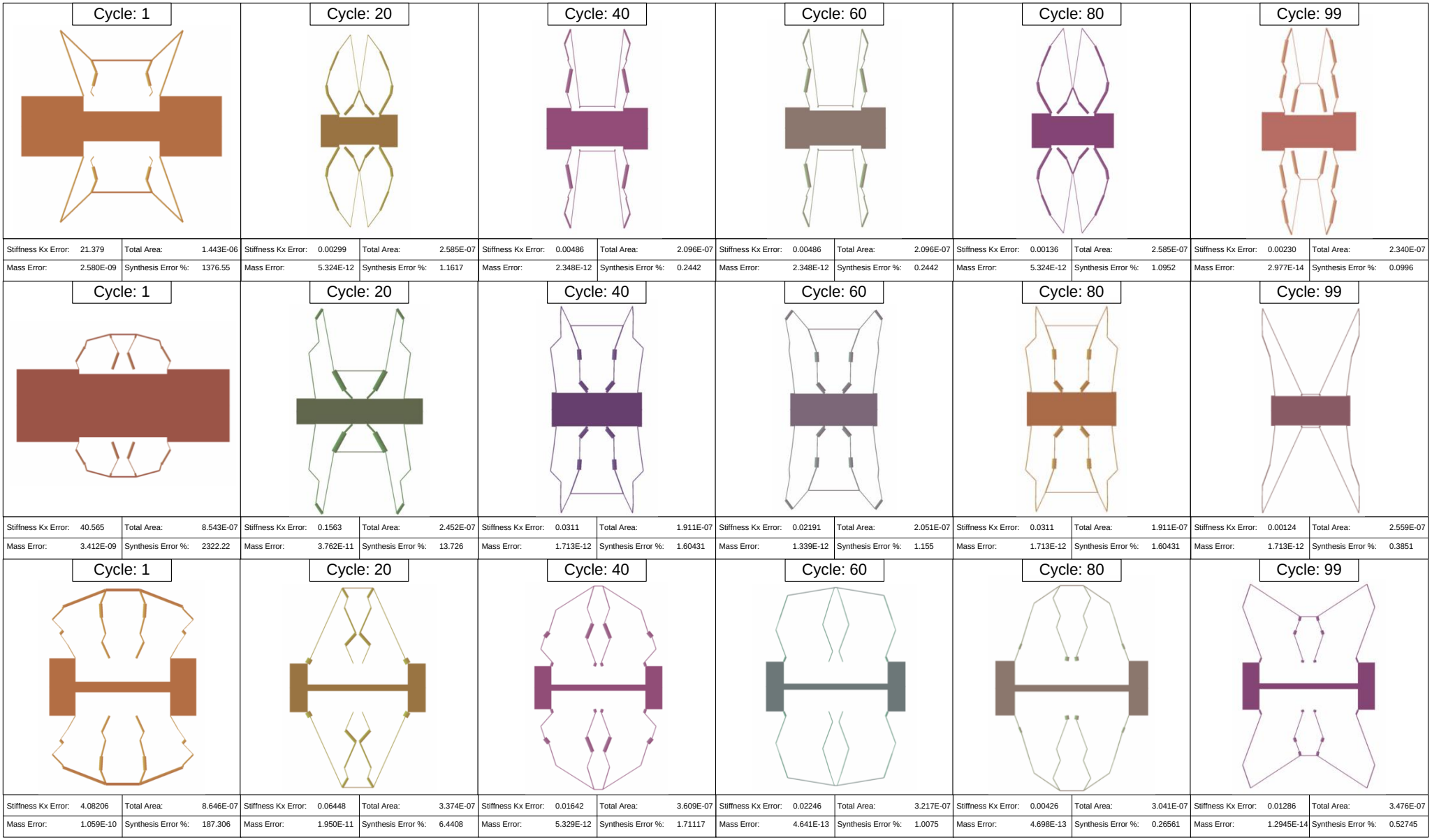
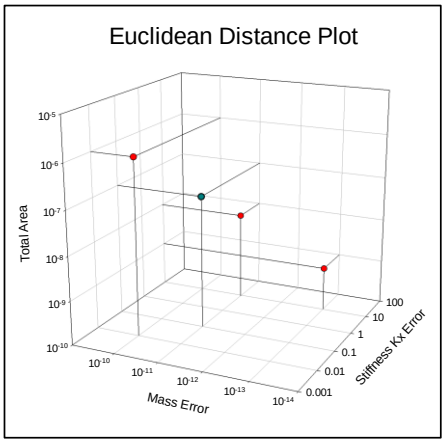


Figure 6.46 Generational plot of the best solution ranked by total synthesis error percentage for device level NSGAII runs 1 (top), 2 (middle) and 5 (bottom)



Solution Objective Values			
Individual	Stiffness Kx Error	Mass Error	Total Area
1	0.01	1E-10	1E-06
2	0.1	1E-13	1E-07
3	10	1E-12	1E-09
4	9	1E-11	1E-08

k^{th} Nearest Neighbour Values				
Individual	1	2	3	4
1		0.090	9.990	8.990
2	0.090		9.900	8.900
3	9.990	9.900		1.0
4	8.990	8.900	1.0	

Figure 6.47 SPEA2 K^{th} nearest neighbour distance plot and values for a three objective problem

Probably the most significant factor which leads to the loss of high quality synthesis objective solutions is the effect the crowding or distance metrics used by both algorithms, have in their component replacement operators. Over time the population will converge to a non-dominated set, where the population members essentially have the same rank or raw fitness value and they have to be evaluated based upon a distance metric. Figure 6.47 gives an example of how a distance metric in SPEA2 works within the three objective folded flexure design problem with the culling of a single solution from a population of four non-dominated individuals. Here individual number two is culled because it has the smallest distance value as a second choice having tied with solution one to begin with. Due to the scale of the stiffness kx error objective with a wide boundary of values (10E-3 to 10E2) in this example, the distance between these objectives for each solution dominates. Therefore solutions which lie close to what is an optimal synthesis value 0.01 will have a short distance and be culled over those which have larger values as seen with values 10 and 9.

The phenotype and its behaviour is also very dynamic, with small changes to value for central mass or the folded flexure giving rise to large variations in there related analysis, be it mass or stiffness. Once the population has converged to a particular phenotype, it can also be hard or impossible for the optimizer to effect enough variation to transverse the design search space to optimal areas for the objectives mass or stiffness. Perhaps the application of multi-level or multidisciplinary optimisation strategies can improve upon this situation.

6.4.2 Multi-level evaluation

The device level of MEMS design optimisation is perhaps the most abstract physical layout level for a MEMS device, often through the use of lumped parameter analytical or NODAL models. Both analytical and NODAL models are relatively quick and reasonably accurate in their analysis with regards to the more costly FEA and BEA analysis associated with the physical level of design [91]. The previous example of multi-level evaluation focused on the system level and circuit analysis tool Spice [55]. In order to define a number of levels which met criteria associated with multi-level evaluation, i.e degrees in cost and accuracy, the Spice analysis tool parameters were varied to give a number of outcomes and levels of accuracy and evaluation cost.

This particular approach is not applicable to any one method as the degree of accuracy cannot be fine tuned in either the analytical or NODAL model of the folded flexure resonator. However it is

possible to use both examples within separate levels themselves with a low cost evaluation occurring using the analytical model and a higher cost NODAL model at a higher level. The difference in functional cost between the analytical and the NODAL analysis calls is shown in table 6.28 giving a rough ratio of ten analytical calls to one NODAL. Though there is a clear distinction in call times between the two methods, it is not clear how each method reflects the accuracy of the real fabricated device and therefore whether the criteria for low and high accuracy can be met. However an important difference between the two modelling methods is the flexibility they possess in relation to the sizing and topology of the MEMS device. The analytical model is stuck to a rigid topological model while the NODAL analysis can be constructed out of a number of atomic elements giving it far more 'creative' freedom, especially with regards to the folded flexure spring component. This freedom and ability to accurately analyse topologies not available in the restricted analytical model make the NODAL model suited to our high level evaluation.

Two migrator modules are utilized in this particular design strategy to allow individuals to move between neighbouring levels. The values for migration percentage along with the cycle count when migration is invoked are shown in table 6.29. The low level pathway is called sequentially throughout the whole design process, while the higher level is called every ten cycles to give a rough 1:1 ratio in terms of functional evaluation cost over the whole design process.

Table 6.28 Evaluation Cost for Analytical and NODAL Folded Flexure Resonator Model

Analytical Analysis	Sugar Analysis
Mean Analysis Time (20 Calls)	Mean Analysis Time (20 Calls)
0.00646037	0.07442126
Ratio	
1 : 11.5196591	

Table 6.29 Migrator Module Parameters for Multi-Level Evaluation

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	20	30
Level 2	Level 1	20	30

A new 'subtree converter' module is added to the design process to aid migration of individuals between the two modelling levels. Conversion from the high level to the low level involves converting the NODAL model to the analytical model outlined in figure 6.48. The spring structures in the NODAL model are converted to the rigid analytical model, afterwards any variables not associated with the NODAL model representation are removed. The beam length, width and truss length variables are added and converted from values from the NODAL model. The low level to high level conversion follows a similar process, with the addition of spring subtree structures, each of which is made up of a random number of beams with a length equally divided between the previous beam length value. An anchor placement length variable is added and based upon the truss length of the analytical model, while the central mass length is also based upon this value and is simply three times the truss length.

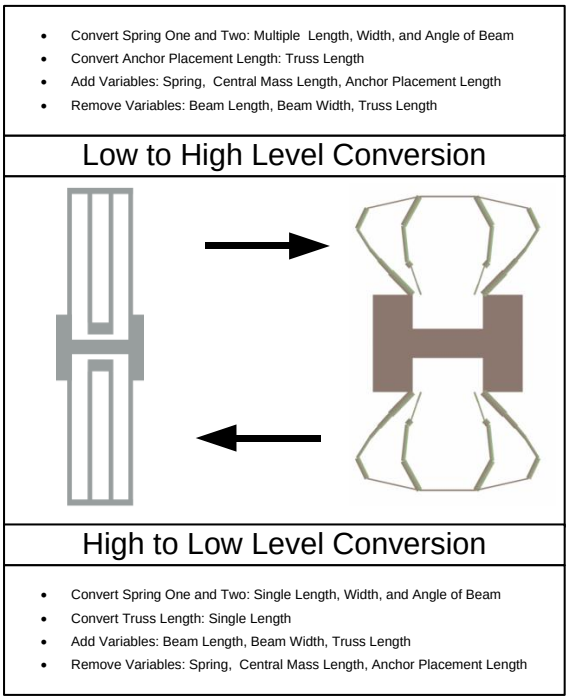


Figure 6.48 Multi level evaluation folded flexure resonator model conversion

The overall multi-level evaluation design process is shown in figure 6.49 and contains both low and high level representations and their associated variables, structure tags and node markers. The NSGAII and SPEA2 algorithm parameters remain the same as the single level strategy with the total cost at 10,000 functional evaluations. The representation parameters are unchanged for the high level NODAL model while the low level representation parameters, objectives and constraints are shown in table 6.30.

Table 6.30 Device Multi-Level Evaluation Level One Folded Flexure Parameters

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Central Mass Width (μm)	Real Valued	10	400
Shuttle Yolk Length (μm)	Real Valued	2	400
Shuttle Yolk Width (μm)	Real Valued	10	400
Beam Length (μm)	Real Valued	5	100
Beam Width (μm)	Real Valued	2	20
Truss Length (μm)	Real Valued	0	400
Truss Width (μm)	Real Valued	2	20
Thickness (μm)	Real Valued	2	200
Objectives		Constraints	
Folded Flexure Stiffness Kx Error	Minimize	Constraint 1	Inequality > 0
Folded Flexure Mass Error	Minimize	[Shuttle Yolk Length - Central Mass Width]	
Total Area	Minimize	Constraint 2	Inequality < 0
		[(([Beam Width] * 4) - ([Truss Length] * 3)) + 9e-6]	

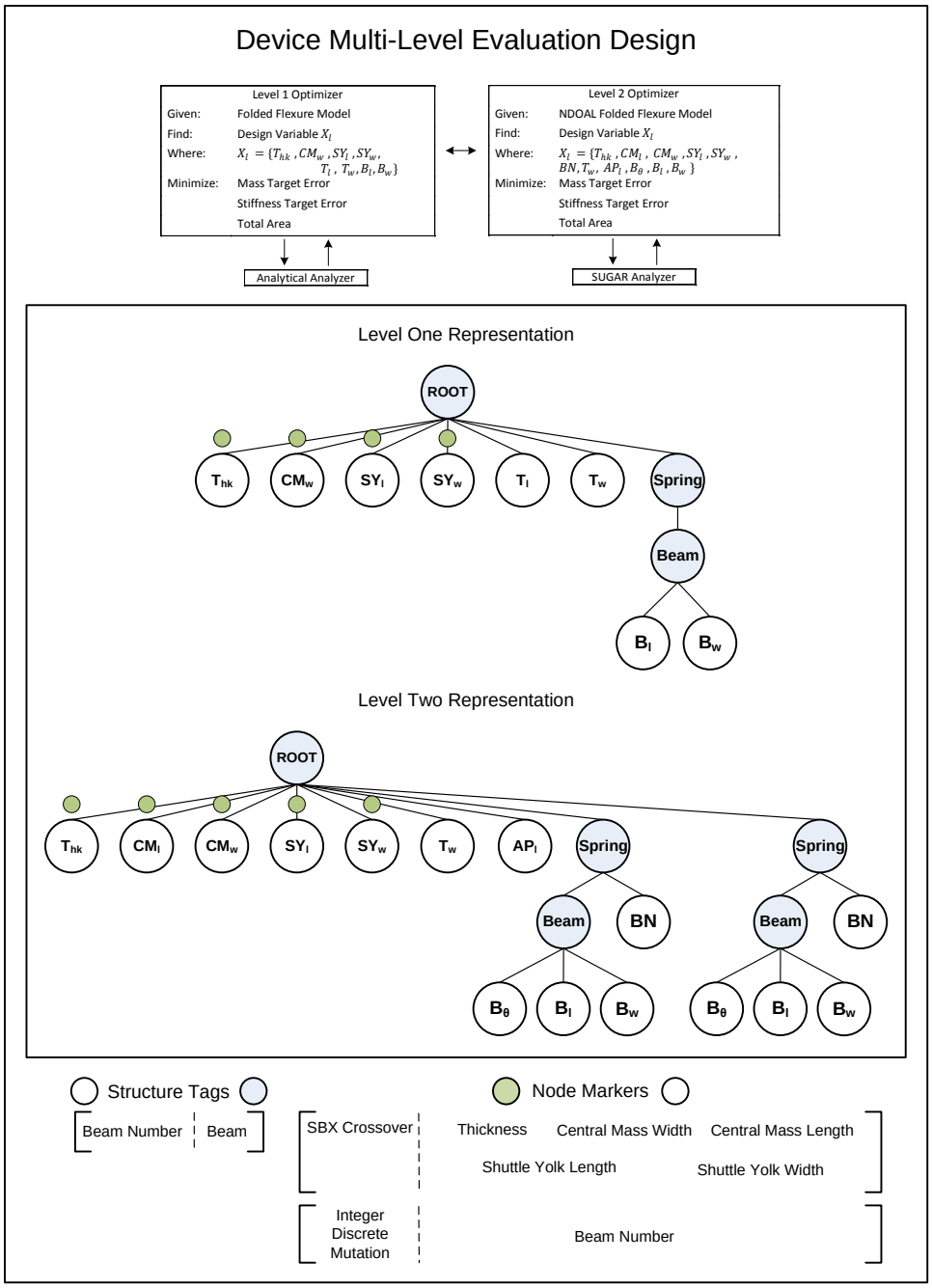


Figure 6.49 Device Multi-Level Evaluation design template, with overview of problem, default representation, associated structure tags and node markers.

6.4.3 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.50 and fully in appendix C.2. A list of the best results ranked by total area and constrained to having a stiffness and mass error of less than 1% from the target for each run are shown in table 6.32 and their phenotypes shown in figure 6.49. Finally the hypervolume values for both algorithms are shown in table 6.31, with the best results shaded.

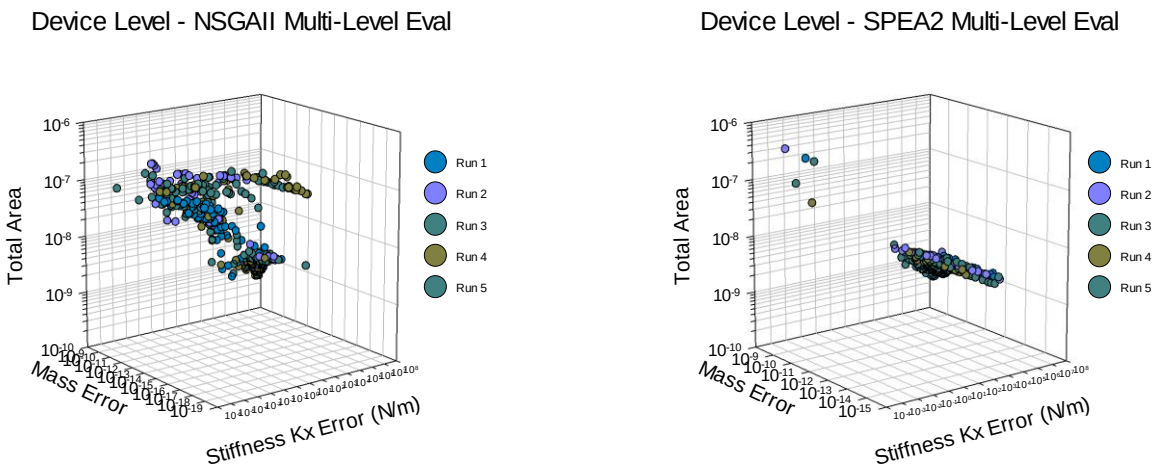


Figure 6.50 Device level run 1 – 5 final population sets for (left) NSGAII multi-level evaluation and (right) SPEA2 multi-level evaluation

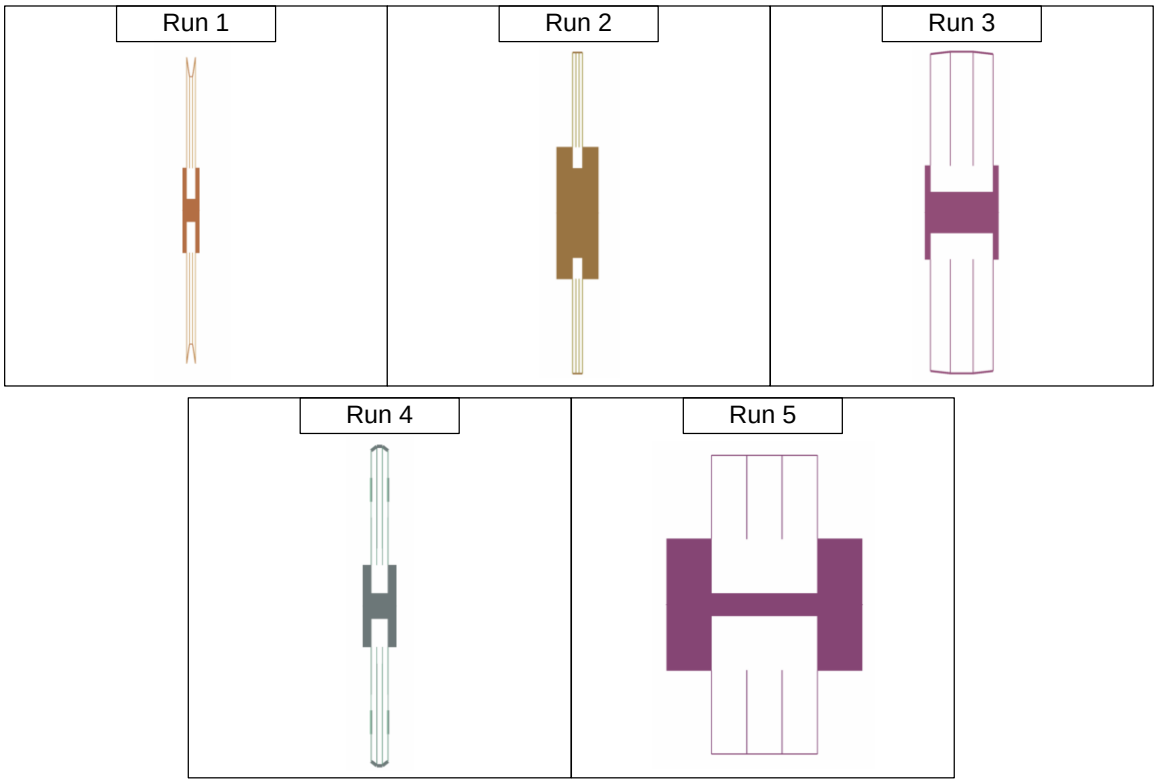


Figure 6.51 Device level run 1 – 5 folded flexure best result ranked by total area for NSGAII multi-level evaluation

Table 6.31 Device Multi-Level Evaluation Hypervolume Results for NSGAII and SPEA2

Device Multi-Level Evaluation		
Hypervolume	NSGAII	SPEA2
S^U	1.29985833605521e-07	1.29932915126865e-07
S^M	1.2997164E-07	1.2958132E-07
S^L	1.29942597914e-07	1.28742002912404e-07

* (S^U S^M S^L) [$1.3e^7$, $1e^{-9}$, $1e^{-5}$]

Table 6.32 Device Multi-Level Evaluation Best Results Ranked by Total Area for Solutions with ≤ 1% Error

Device Multi-Level Evaluation NSGAII					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	Mass Error	Total Area
1	100	3	0.00072376	2.1353305E-14	5.0162732E-08
2	100	5	0.01507497	1.9379714E-13	1.0551121E-07
3	100	2	0.013156629	2.8961064E-15	1.0748359E-07
4	100	3	0.000893577	1.8418389E-12	8.3184375E-08
5	100	2	0.00019868	8.6060228E-16	2.0917228E-07
Device Multi-Level Evaluation SPEA2					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	Mass Error	Total Area
1	100	0	Not Applicable		
2	100	0			
3	100	0			
4	100	0			
5	100	0			

The coupling of two device level folded flexure resonator models in a multi-level evaluation strategy has proven successful in the task of synthesizing and optimising the device to match design target and cost goals. Once again SPEA2 fails to find solutions which have a synthesis error of less than 1 percent and mirroring the single level strategy by focusing heavily on the total area objective. There is a slight increase in the number of solutions found within each run which have a synthesis error less than 1 % for NSGAII compared with the single level strategy. The hypervolume results once again show NSGAII to be the superior of the two algorithms performing on average better and giving the best result of all the runs. Figure 6.52 compares the generational hypervolume values of the multi-level evaluation strategies with the single level methods for both NSGAII and SPEA2, and in both cases there is an improvement in performance, significantly it would seem with SPEA2 multi-level evaluation. Both algorithms for the multi-level strategy exhibit a fast growth in hypervolume performance before an abrupt stalling and flattening of the improvement covering the majority of the design process.

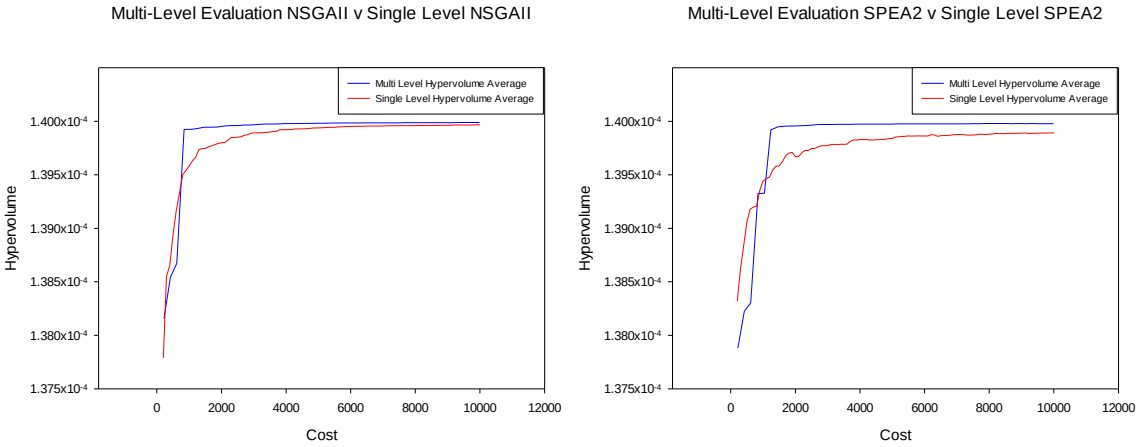


Figure 6.52 Device level average hypervolume results for the 5 runs of the multi-level evaluation and single level NSGAII and SPEA2 strategies * (S^U S^M S^L) [1.4e7, 1e-6, 1e-5].

One of the major additional factors that have influenced the performance of the multi-level strategy is the low level analytical model itself and its rigid topology and reduced variables. Looking over figure 6.51 it is clear that this topology is present or has influenced the best solutions found within each run of the NSGAII multi-level evaluation strategy. The additional functional evaluations as a result of the multi-level evaluation strategy than provide the necessary exploration of this topology to give a folded flexure device which match target goals and have a very small total area footprint.

6.4.4 Multi-level parameterization

The standard topology of the folded flexure resonator is akin to the analytical model with a simple single beam construction for each spring of the folded flexure attached to a suspended central mass component. The NODAL model expands this particular topology to allow for more complex flexure structures made up of a number of individual beam elements, each of which contains its own length, width and angle values. The sequential coupling of these beams can give rise to numerous designs often with imaginative shapes as seen in the single level design strategy. However with each additional beam comes the potential cost of increased complexity within the design search space as the optimizer has to handle and evolve more decision variables and the potential for constrained designs increases. Therefore it could be beneficial to allow the optimizer to evolve simplified representations based upon the standard folded flexure topology and separately evolve increasingly more complex representations elsewhere. The device multi-level parameterization strategy focuses upon this be including a number of levels each containing representations of varying parameterization, and a migration policy which transfers solutions between such levels.

Table 6.33 contains the variable parameters for each of the three levels and this is reflected in the folded flexure component within figure 6.53. Here level one has the simplest model representation, with only one spring being evolved and cloned for all others, the spring consists of multiple beams but only the first of which is evolved while all others are clones, there is also a restricted parameterization with a fixed angle and the anchor placement is also fixed. The second level loosens the representation to now allow each beam to have its own width and a varied anchor placement length. The final third level is unrestricted and simply mirrors the standard single level representation. The effect each parameterization method has on the number of decision variables is shown in table 6.34.

Four migrator modules are utilized in this particular design strategy to allow individuals to move between neighbouring levels. The values for migration percentage along with the cycle count when migration is invoked are shown in table 6.35. All of the pathways are called sequentially throughout the whole design process and as a result this gives around 3000 to 3100 functional evaluations per level.

The overall multi-level parameterization design process is shown in figure 6.54 and contains all of the parameterization representations and their associated variables, structure tags, global and node markers. The NSGAII and SPEA2 algorithm parameters remain the same as the single level strategy with the total cost at 10,000 functional evaluations.

Table 6.33 Device Multi-Level Parameterization Filter Problem Variable Parameters

		Level One		Level Two		Level Three	
Variable Tag	Sub Tree Type	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound
Central Mass Length (μm)	Real Valued	10	600	10	600	10	600
Central Mass Width (μm)	Real Valued	10	400	10	400	10	400
Shuttle Yolk Length (μm)	Real Valued	2	400	2	400	2	400
Shuttle Yolk Width (μm)	Real Valued	10	400	10	400	10	400
Beam Number	Integer	1	6	1	6	1	6
Beam Angle	Real Valued	90	90	90	90	45	135
Beam Length (μm)	Real Valued	5	100	5	100	5	100
Beam Width (μm)	Real Valued	2	20	2	20	2	20
Truss Width (μm)	Real Valued	2	20	2	20	2	20
Anchor Placement Length (μm)	Real Valued	30	30	0	400	0	400
Thickness ($\boldsymbol{\mu m}$)	Real Valued	2	200	2	200	2	200

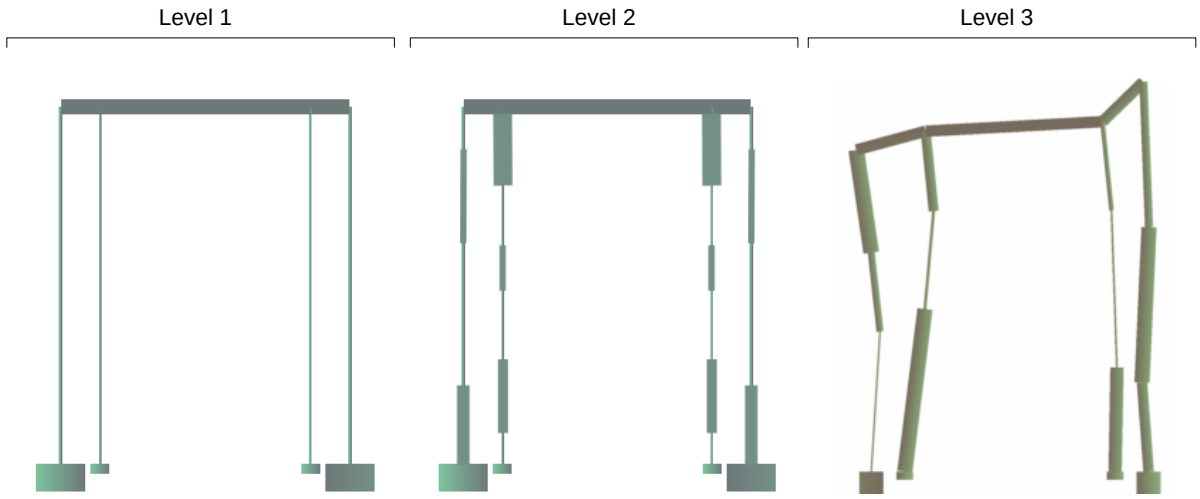


Figure 6.53 Device multi-level parameterization folded flexure phenotypes for levels one (left), two (middle) and three (right).

Table 6.34 Device Multi-Level Parameterization Level Variable Count

	Level One		Level Two		Level Three	
	1 Beam	6 Beams	1 Beam	6 Beams	1 Beam	6 Beams
Variable Count	9	9	10	17	11	26

Table 6.35 Migrator Module Parameters for Multi-Level Parameterization

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	20	4
Level 2	Level 1	20	4
Level 2	Level 3	20	4
Level 4	Level 3	20	4

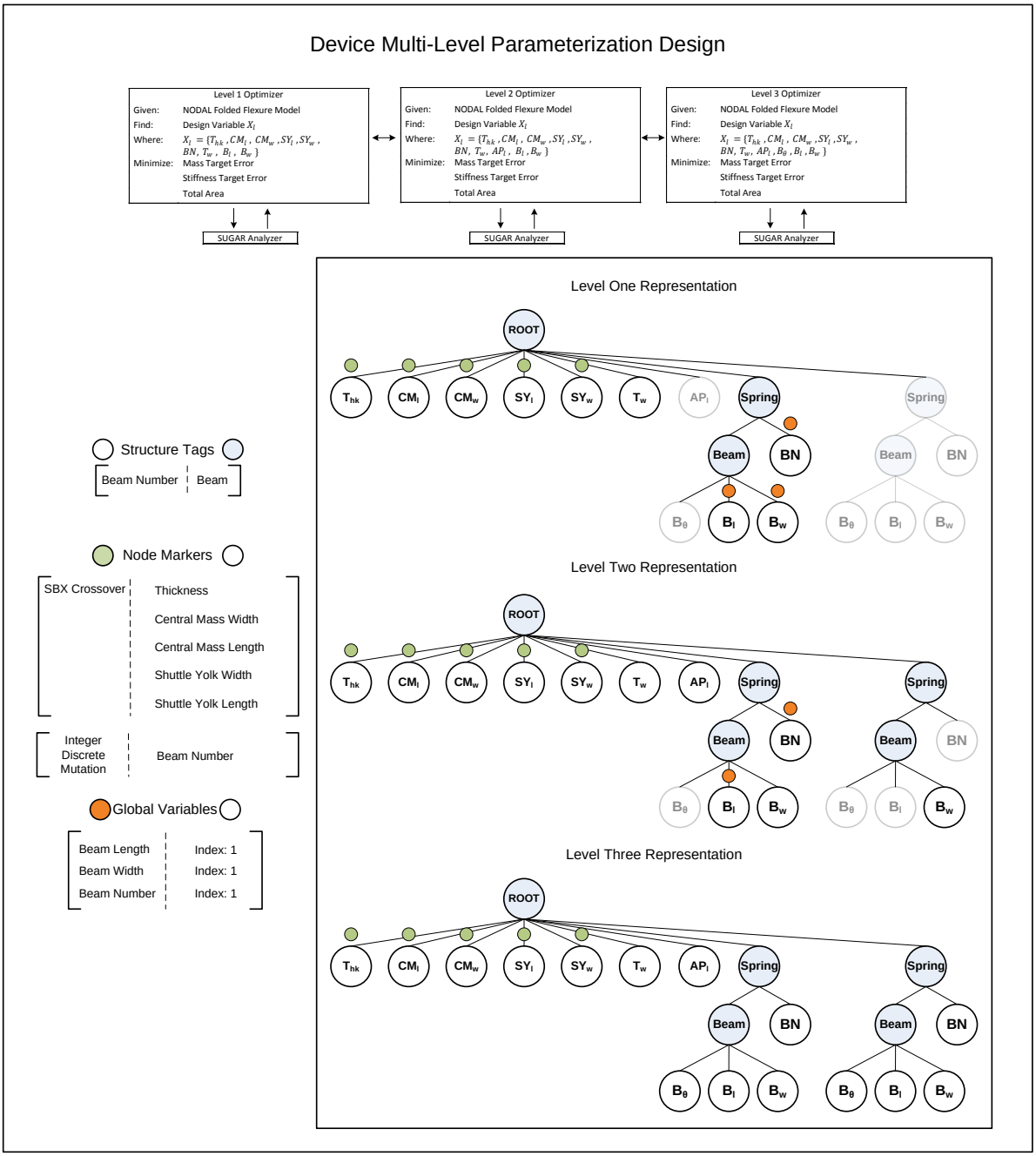


Figure 6.54 Device Multi-Level Parameterization design template, with overview of problem, default representations, associated structure tags, global and node markers.

6.4.5 Numerical Results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.55 and fully in appendix C.2. A list of the best results ranked by total area and constrained to having a stiffness and mass error of less than 1% from the target for each run are shown in table 6.37 and their phenotypes shown in figure 6.56. Finally the hypervolume values for both algorithms are shown in table 6.36, with the best results shaded.

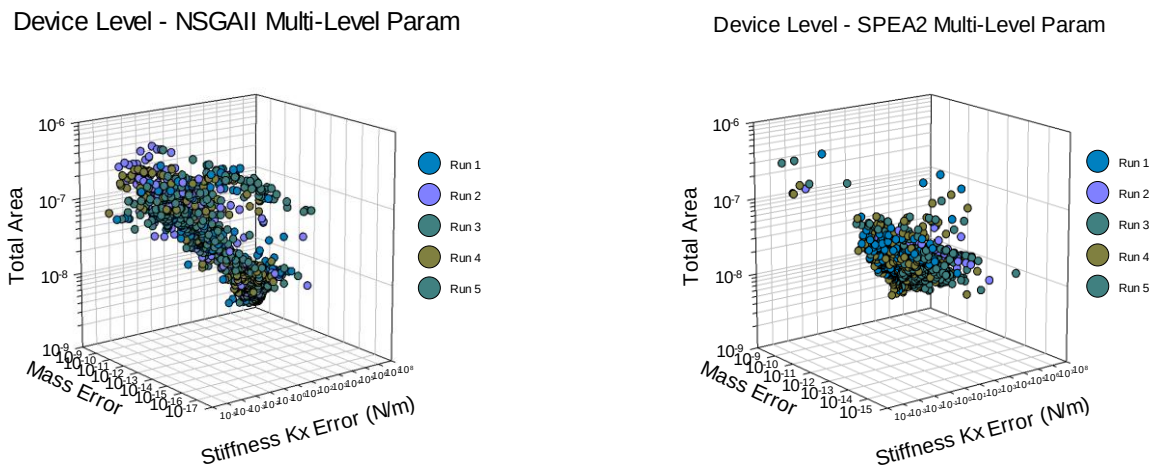


Figure 6.55 Device level run 1 – 5 final population sets for (left) NSGAII multi-level parameterization and (right) SPEA2 multi-level parameterization

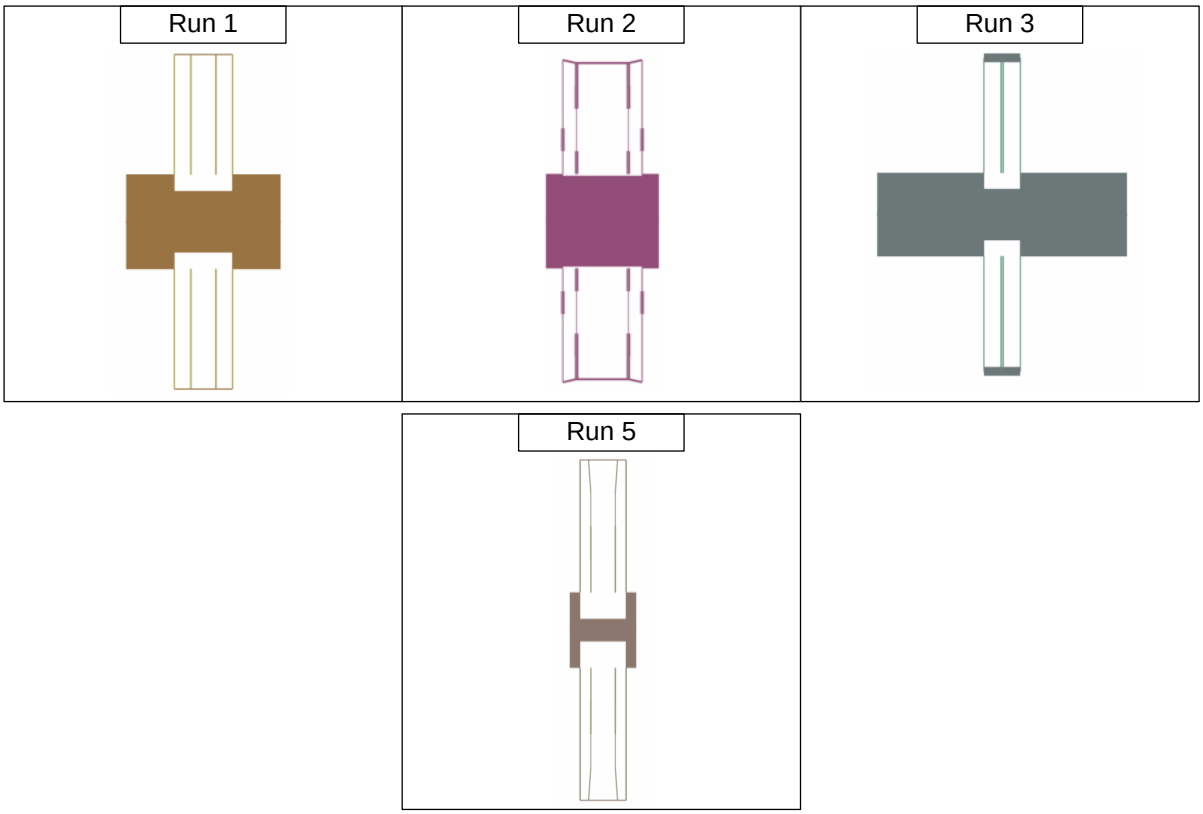


Figure 6.56 Device level run 1 – 5 folded flexure best result ranked by total area for NSGAII multi-level parameterization

Table 6.36 Device Multi-Level Parameterization Hypervolume Results for NSGAII and SPEA2

Device Multi-Level Parameterization		
Hypervolume	NSGAII	SPEA2
S^U	1.29960502201653e-007	1.29935645705392e-007
S^M	1.29953601e-07	1.29914260e-07
S^L	1.29945055166761e-007	1.29899070681417e-007

* (S^U S^M S^L) [$1.3e^7$, $1e^{-9}$, $1e^{-5}$]

Table 6.37 Device Multi-Level Parameterization Best Results Ranked by Total Area for Solutions with ≤ 1% Error

	Device Multi-Level Parameterization NSGAII					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Level	Stiffness Kx Error	Mass Error	Total Area
1	300	2	Hi	0.009004304	3.3576988E-12	1.6792378E-07
2	300	2	Hi	0.0223184675	3.0004161E-13	1.7451476E-07
3	300	3	Hi	0.017257752	2.7740959E-12	2.3764678E-07
4	300	0	-	-	-	-
5	300	1	Hi	0.00953903	2.0093540E-13	1.7711777E-07
	Device Multi-Level Parameterization SPEA2					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Level	Stiffness Kx Error	Mass Error	Total Area
1	300	0	-			
2	300	0	-			
3	300	0	-	Not Applicable		
4	300	0	-			
5	300	0	-			

The multi-level parameterization strategy shows similar performance to other strategies with NSGAII outperforming SPEA2 though failing slightly on one experiment to find a solution with a synthesis error of less than 1%. In comparison with the single level strategy the multi-level NSGAII approach converges to similar final populations, though the effect of additional levels of parameterization alter performance characteristics early in the design process. However even though the final population sets between single and multi-level NSGAII are similar on average, the multi-level parameterization strategy is able to find solutions that outperform the single level in terms of total area cost at a 1% synthesis error or less. In regards to the SPEA2 algorithm, the multi-level parameterization strategy shows a marked increase in final solution hypervolume performance though it is still unable to find solutions with a synthesis error lower than 1%. The addition of multiple levels does have an added bonus in providing more Pareto solutions and more variation in choice of design which can be presented to the designer, along with a number of parameterization templates which give unique topologies to choose from.

Comparison of the multi-level parameterization strategy generational hypervolume performance to that of the single level are shown in figure 6.57 for both NSGAII and SPEA2 algorithms. Each of the three parameterization levels and their associated population sets are represented as individual plots. Both the single and multi-level NSGAII methods show similar performance while for SPEA2 the multi-level method is superior overall. There are also characteristic differences between the performances of each level within the multi-level parameterization strategy. The largest gain in performance is met early on by level one, the simplest parameterization before migration operators are able to pass this gain to the other levels. As the design process reaches the end of the experimental run it can be seen that the performance has switched with level three and two providing the larger hypervolume performance gain followed by one, though for NSGAII this divide is not as great.

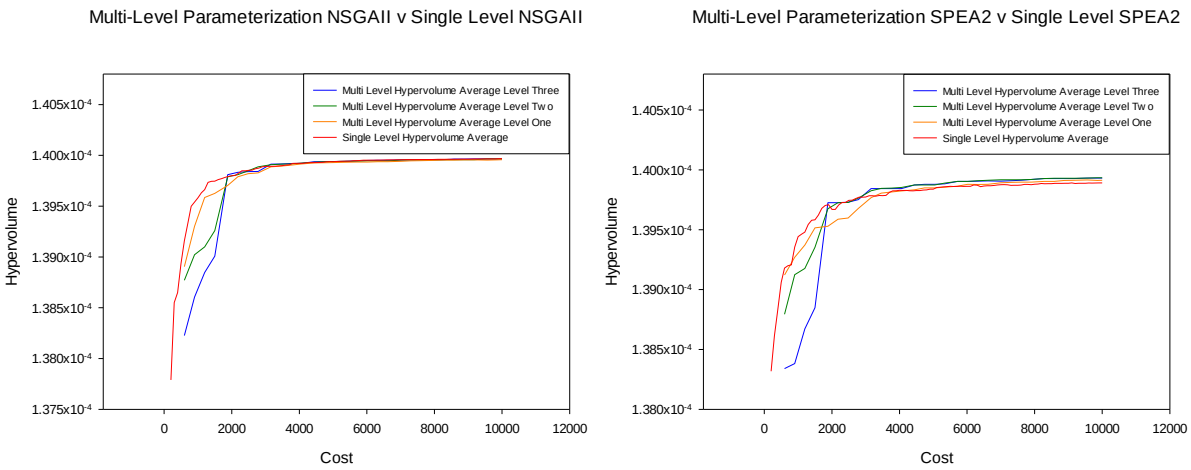


Figure 6.57 Device level average hypervolume results for the 5 runs of the multi-level parameterization and single level NSGAII and SPEA2 strategies * (s^U s^M s^L) [1.4e7, 1e-6, 1e-5].

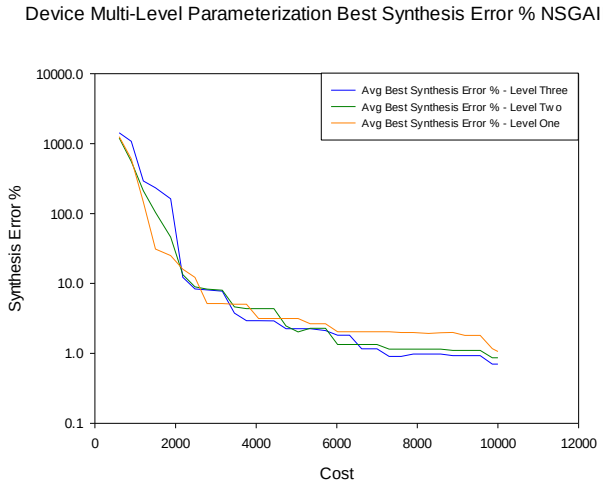


Figure 6.58 Device level average best synthesis error % results for 5 runs of multi-level parameterization NSGAII levels one, two and three.

Looking at the synthesis error for the best solutions found at each level over the design process shows a similar characteristic to that of the hypervolume performance with level one evolving better solutions early on before levels two and three begin to perform better and over take performance at the end as seen in figure 6.58. The simplicity of the level one parameterization, in particular with solution initialisation may help in providing solutions which can be evolved more readily towards improved synthesis objective values. As the design process progresses these solutions are distributed between the other levels where the increased exploration as a result of their relaxed parameterization schemes can allow for further improvement in synthesis error reduction.

The multi-level parameterization strategy allows for evolved topologies to be transferred from one level to another and utilize their specific representations to aid the design process to improve performance for the synthesis objectives. Figure 6.59 shows a generational plot of the best synthesis error % solutions found over the design process for each level of NSGAII run 1.

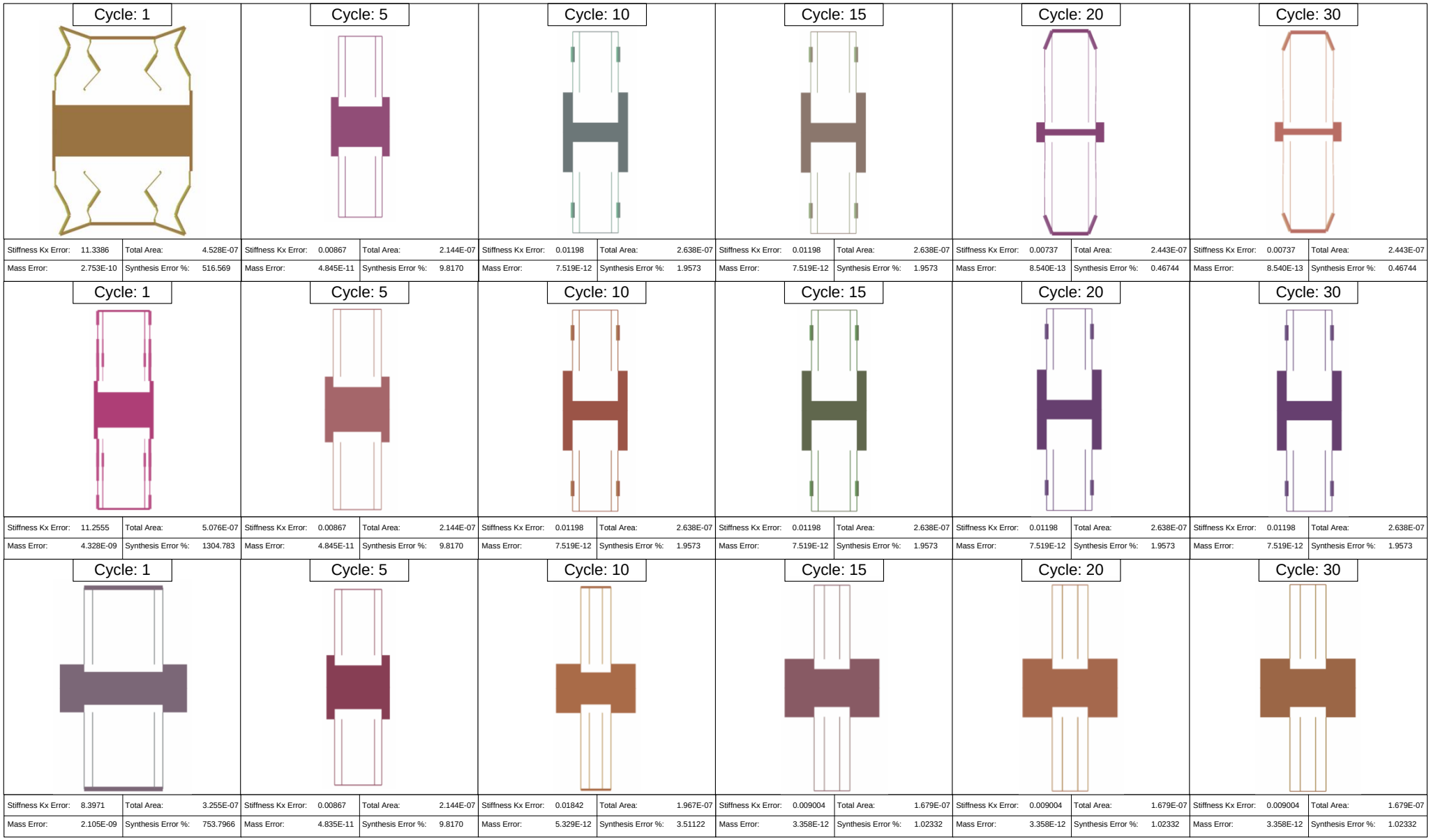


Figure 6.59 Generational plot of the best solution ranked by total synthesis error percentage for device multi-level parameterization NSGAII run 1 - level three (top), level two (middle) and level one (bottom).

Here solutions evolve over the design process and each of the levels contributes in some way towards the best synthesis error solution. The first solution to be adopted over all three levels occurs at cycle 5 from level one of the multi-level strategy. This particular solution is altered at the second level with a change to the central mass width and various individual beam thickness values and passed on to level three at cycle 10. It is from here that levels one and two retain a solution and leave it relatively unchanged until the end of the design process with both having synthesis error values near the target of 1%. The unrestricted parameterization of level three however is able to evolve a solution which outperforms all others and with a synthesis error of less than 1% through alteration of the central mass, shuttle yolk and various beam angles specific to this levels parameterization.

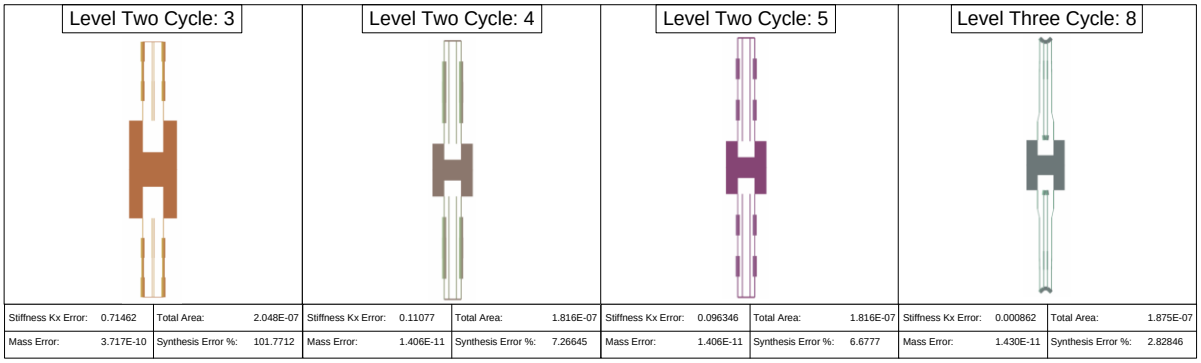


Figure 6.60 Generational plot of the best solution ranked by total synthesis error percentage for device multi-level parameterization NSGAII run 3 for levels one and two.

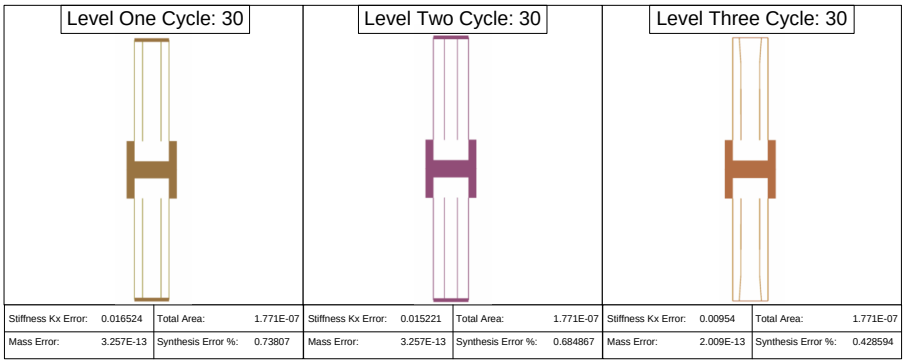


Figure 6.61 Generational plot of the best solution ranked by total synthesis error percentage for device multi-level parameterization NSGAII run 5 for levels one, two and three.

The ability for gradual variation in a step wise manner from restricted to unrestricted parameterization levels has clear benefits in improving performance, in particular with those objectives associated with the synthesis error. Figure 6.60 shows another example early on in the design process of level twos variation of a particular solution over a number of cycles and the associated improvement in synthesis error and how this solution is altered more freely at level three to give a significant improvement over the previous designs. The differences in each levels parameterization and their effect on synthesis error at the end of the design process are shown in figure 6.61 for NSGAII run 5. The transition from the level one design to level two results in a slight variation in the anchor placement length variable which is specific to this level,

and from level two to level three a variation in beam angle of the spring two all of which gives rise to an incremental improvement in synthesis error performance.

6.4.6 Multidisciplinary optimisation

The MEMS folded flexure resonator can be split into two separate functional components, the folded flexure springs and the central mass as seen in figure 6.62. These components play the largest part in both the stiffness kx error and mass error objectives outlined previously.

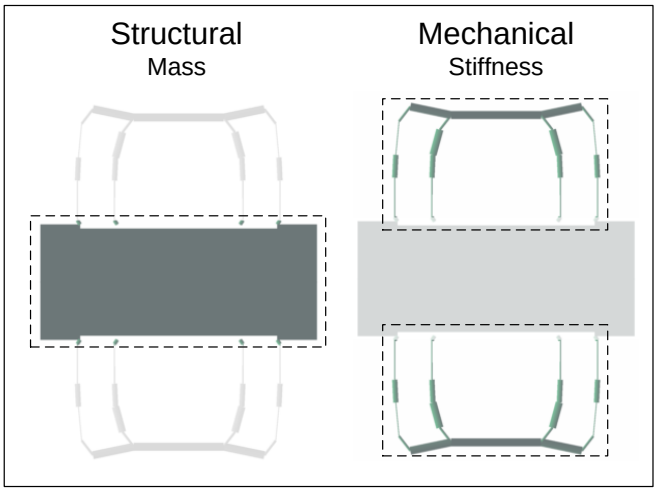


Figure 6.62 MEMS folded flexure resonator functional components for structural discipline (mass) and mechanical discipline (stiffness).

The principles of multidisciplinary optimisation based upon object based decomposition allow for separate optimizers to focus upon each of the functional components and their associated synthesis and cost objectives. Each separate subsystem focusing upon the decision variables associated with each functional component and local selection pressure associated with the functional objectives could benefit the whole design process by decomposing the problem into a number of smaller tasks.

The multidisciplinary optimisation strategy is outlined in figure 6.63 and consists of a single system and two subsystem pathways. Their associated representations are included with subsystem one containing decision variables related to the mass objective while subsystem two contains variables related to the folded flexure component and the stiffness objective. The system level often contain decision variables that are applicable to more than one subsystem, such as thickness or anchor placement length both of which have a direct influence on the mass and stiffness subsystems. A decision was taken based upon prior experimentation and analysis to place the thickness variable within the mass subsystem as its role was heavily influential to the objectives within it and placing the variable within the system representation meant it was not evolved sufficiently over the design process. The anchor placement length has an effect on the topology of the central mass component, in particular the central mass length, with resulting conflict between both variables arising as a result of any undesired spring intersection. Anchor placement length also affects the overall topology of the folded flexure component and therefore the decision variable is placed within the system level so it is removed from any direct control from the subsystems optimizers. Also included in figure 6.63 are the structure and node

markers for each representation, with system and subsystem markers indicating what variables can be varied in their associated pathways.

In addition to the changes to the representation for each separate system or subsystem pathway is a change in the objectives used. The default objectives found within the single level strategy are also found within the system level with both synthesis objectives mass and stiffness error, and cost objective total area used. Subsystem one simply contains the mass and total area objectives while subsystem two focuses upon the stiffness error and total area objectives as seen in table 6.38. An additional constraint is added to subsystem two to restrict the stiffness error to be less than 100% of the target.

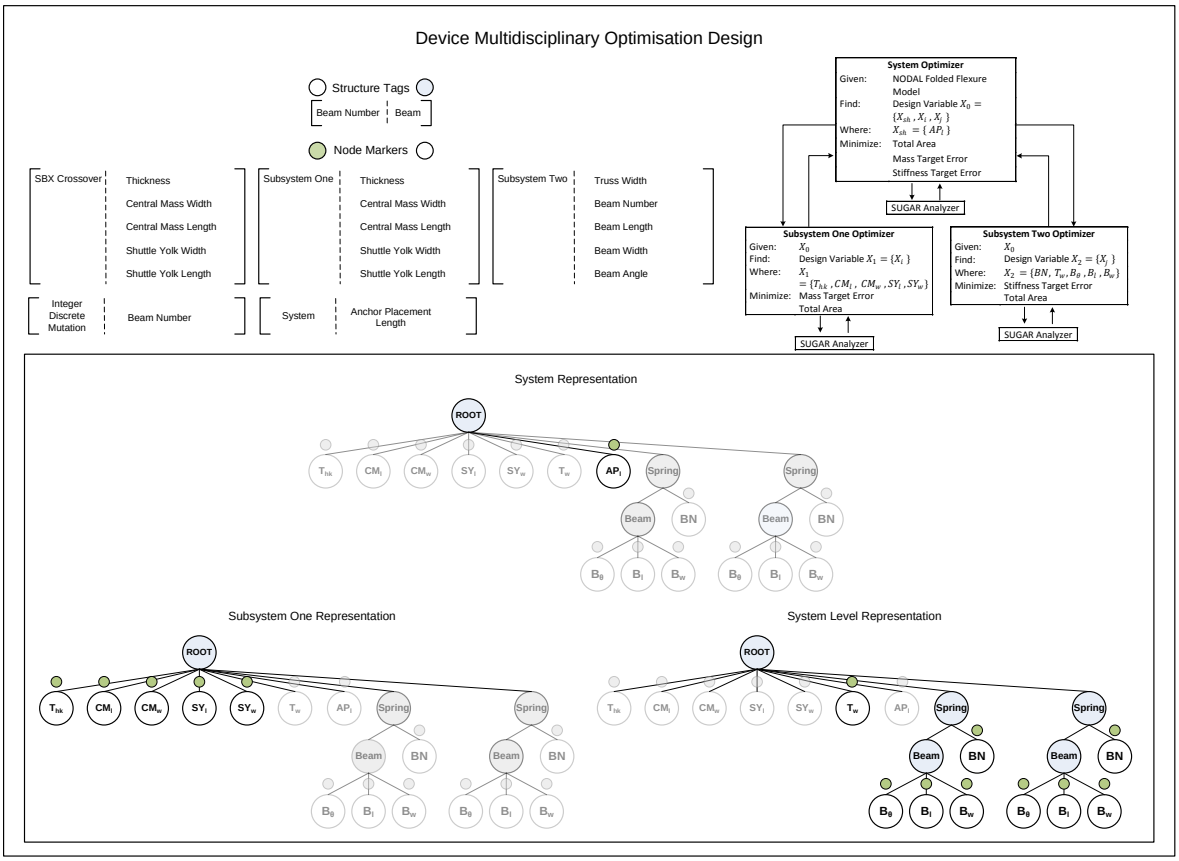


Figure 6.63 Device multidisciplinary optimisation design template, with overview of problem, default representations, associated structure tags, and node markers.

Table 6.38 Device Level Multidisciplinary Optimisation Objectives

	System Level			Subsystem 1		Subsystem 2	
	Objective 1	Objective 2	Objective 3	Objective 1	Objective 2	Objective 1	Objective 2
Objective Type	Maximize	Minimize	Minimize	Minimize	Minimize	Minimize	Maximize
Objective Description	Total Area	Stiffness Kx Error	Mass Error	Mass Error	Total Area	Stiffness Kx Error	Total Area
Constraint Type	Inequality > 0			Inequality		Inequality < 100%	
Constraint Description	Constraint 1 [Shuttle Yolk Length - Central Mass Width]			Intersection Check		Constraint 2 (Subsystem Two) [Stiffness Kx Error]	

The NSGAII and SPEA2 algorithm parameters remain the same as the single level strategy with the total cost at 10,000 functional evaluations. This is split between the system and subsystem pathways, with the system level pathway being called every 10 cycles while the subsystem pathways are called sequentially over the design process.

6.4.7 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.64 and fully in appendix C.2. A list of the best results ranked by total area and constrained to having a stiffness and mass error of less than 1% from the target for each run are shown in table 6.39 and their phenotypes shown in figures 6.65 and 6.66. Finally the hypervolume values for both algorithms are shown in table 6.40, with the best results shaded.

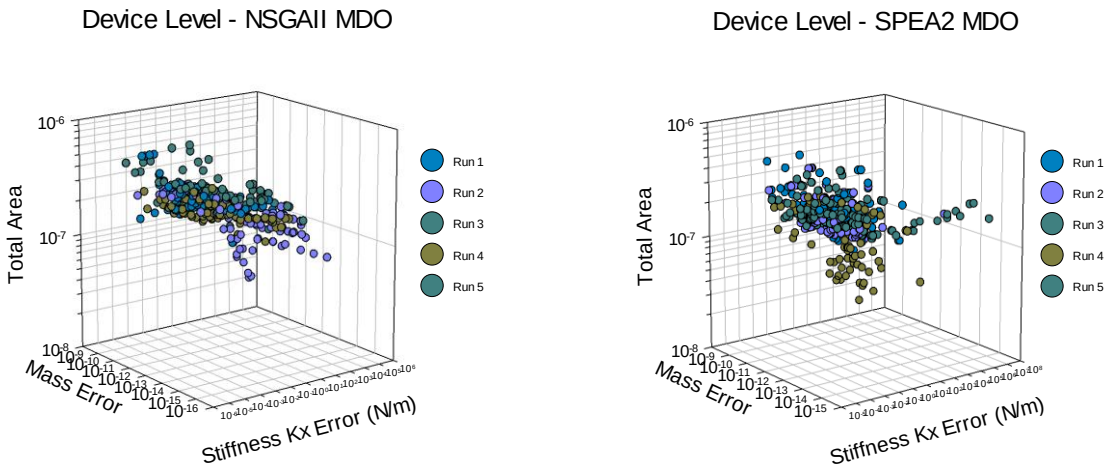


Figure 6.64 Device level run 1 – 5 final population sets for (left) NSGAII multidisciplinary optimisation and (right) SPEA2 multidisciplinary optimisation

The application of a multidisciplinary optimisation strategy to the design optimisation of a device level folded flexure resonator has been undertaken with a stark change in the outcome of the final population sets compared with previous strategies. Looking at both the NSGAII MDO and SPEA2 MDO population set results in figure 6.64, the trend for the optimizers to focus heavily on solutions with a small total area and as a consequence often a large synthesis error is not present. Instead the majority of the solutions lie above $10e^{-7}$ total area, spanning across both mass and stiffness error objectives though at a reduction in error, in particular with stiffness kx error.

The effect of the MDO strategy on the number of solutions with a synthesis error < 1% in table 6.39 has led to an increase with a larger percentage of solutions having this low synthesis error for NSGAII while the SPEA2 algorithm was also finally able to find a solution with an error < 1%. This reduction however is not coupled with a significant reduction in total area and compared with the single level strategy the solutions themselves cover a larger surface area than the single level solutions. The lack of solutions with a reduced total area has had an effect on the overall hypervolume results for the MDO strategy as a whole and each of the NSGAII and SPEA2 algorithms as seen in table 6.40 and figure 6.67 of the generational plots for the average

hypervolume for MDO and single level strategies. The hypervolume values are representative of the area of the objective space that is dominated by the final Pareto sets and the chosen nadir point. All previous strategies gave rise to a number of solutions with a low total area, near $10e^{-9}$, where as the MDO strategy contained solutions mainly around $10e^{-7}$. The chosen nadir point may also play an effect on which of the solutions objective values contribute greater to the hypervolume value and with the lack of solutions with a low total area, retard the hypervolume performance overall.

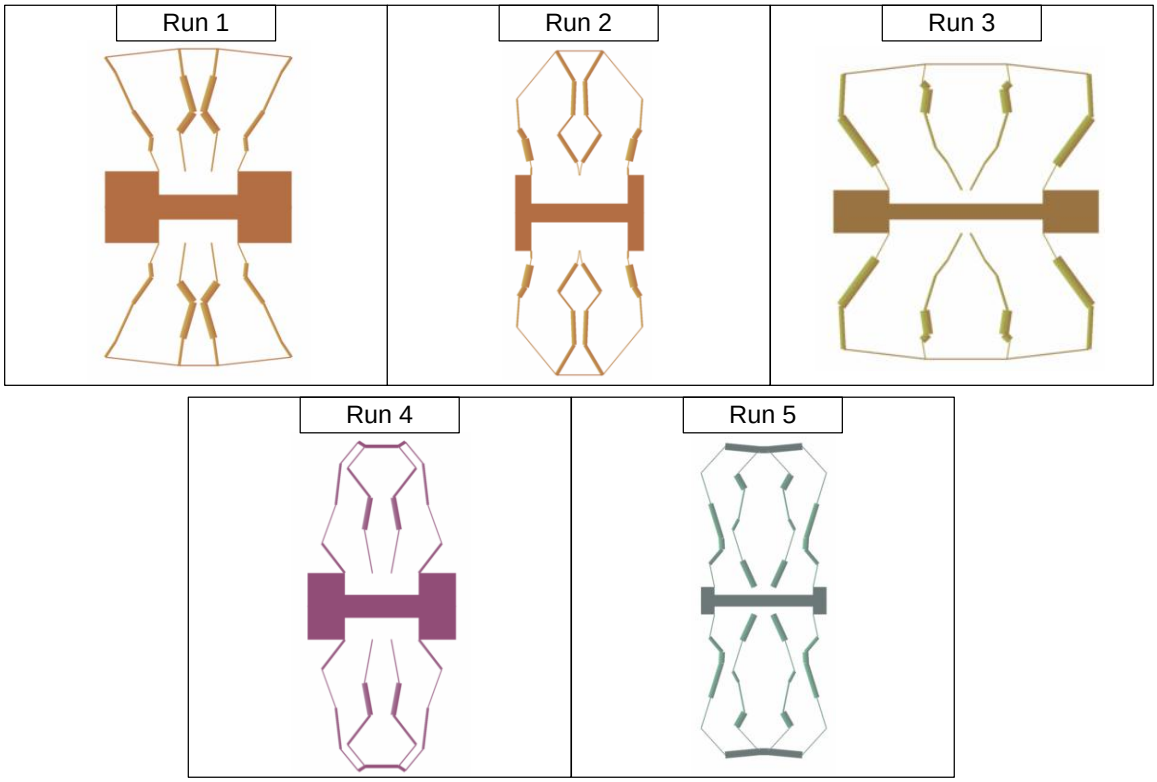


Figure 6.65 Device level run 1 – 5 folded flexure best results ranked by total area for NSGAII multidisciplinary optimization - object Based

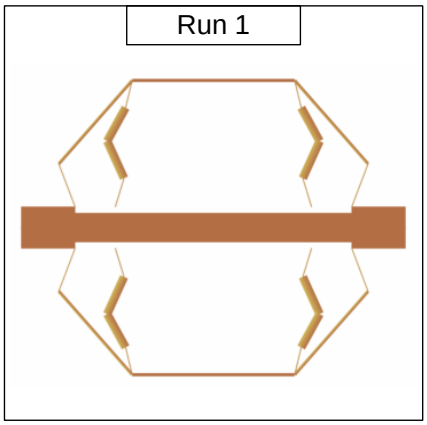


Figure 6.66 Device level run 1 folded flexure best result ranked by total area for SPEA2 multidisciplinary optimization - object Based

Table 6.39 Device Level Multidisciplinary Optimisation Best Results Ranked by Total Area for Solutions with ≤ 1% Error

Device Multidisciplinary Optimization NSGAII					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	Mass Error	Total Area
1	91	3	0.009576541	3.40620516E-14	2.5130852E-07
2	100	4	0.00937402	3.6965257E-12	2.590366E-07
3	72	12	0.0175072557	1.13987359E-13	2.9488664E-07
4	96	9	0.000951855	3.06808698E-12	2.1144589E-07
5	98	15	0.023429797	2.22778495E-12	2.8347466E-07
Device Multidisciplinary Optimization SPEA2					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	Mass Error	Total Area
1	100	1	0.021473	3.32827E-12	4.721381E-07
2	100	0			
3	100	0			
4	100	0			
5	100	0			

Not Applicable

Table 6.40 Device Level Multidisciplinary Optimisation Hypervolume Results for NSGAII and SPEA2

Device Multidisciplinary Optimisation		
Hypervolume	NSGAII	SPEA2
S^U	1.29553948363743e-007	1.2970688951989e-007
S^M	1.2898315E-07	1.2902204E-07
S^L	1.28466481703699e-007	1.28572875228194e-007

* (S^U S^M S^L) [$1.3e^7$, $1e^{-9}$, $1e^{-5}$]

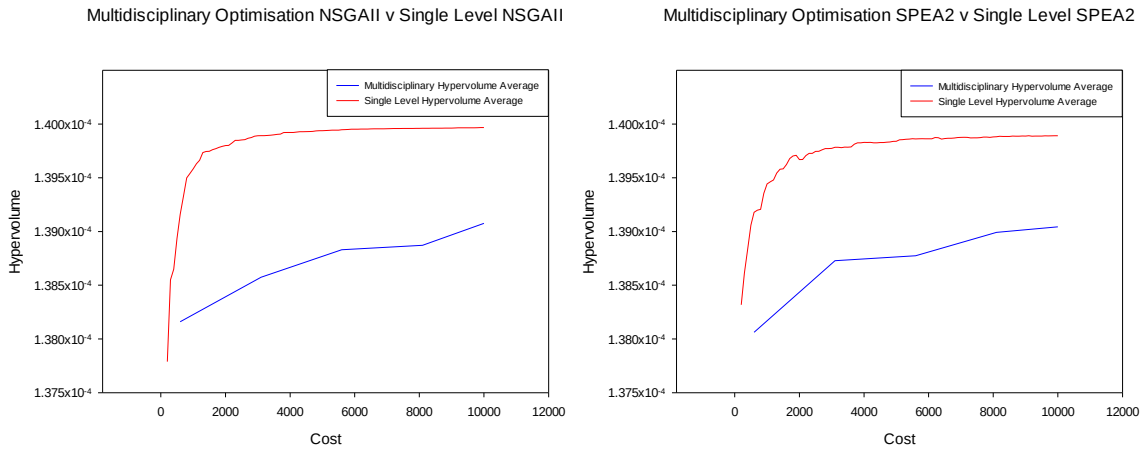


Figure 6.67 Device level average hypervolume results for the 5 runs of the multidisciplinary optimisation and single level NSGAII and SPEA2 strategies * (S^U S^M S^L) [1.4e7, 1e-6, 1e-5].

The main focus of the multidisciplinary strategy was to aid the optimisation of the mass and stiffness kx error objectives through the design synthesis of the functional components of the central mass and folded flexure springs of the resonator device. The partitioning of the device into a number of subsystems to allow the evolution of designs with reduced synthesis error that could simultaneously be optimised in their total area could lead to an improved overall performance in the design optimisation of the folded flexure resonator.

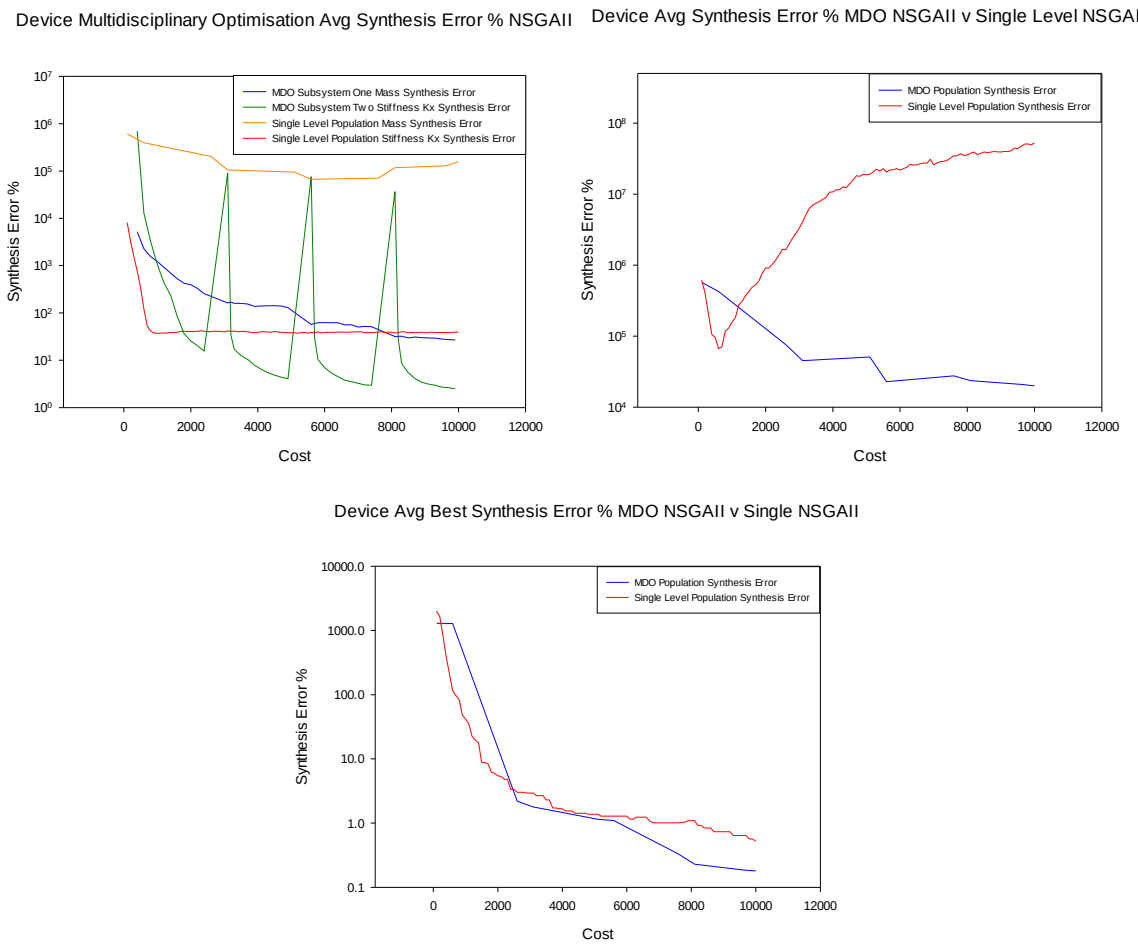


Figure 6.68 Device level multidisciplinary optimisation analysis plots for average synthesis error % for mass and stiffness kx error objectives (top left), average total synthesis error % (top right), and average best synthesis error % (bottom) for single level NSGAII and MDO NSGAII runs 1 – 5.

Looking at each subsystem individually and their effect on the mass and stiffness kx error synthesis objectives and comparing them with the single level approach in figure 6.68 there is a clear distinction between both strategies. The MDO strategy focuses the optimisation of each synthesis objective in a separate subsystem, with the gradual reduction of the total average synthesis error % for mass while the single level strategy converges very fast to a value around 40% total error for the population sets on average. Though it takes longer for the MDO strategy to reach this level of reduced total synthesis error it is able to improve upon it reaching around 20-25% total error. The stiffness synthesis error for subsystem two of the MDO strategy and the single level population have the widest discrepancy between performance, with the MDO strategy able to drive the total synthesis error to around 2% while the single level strategy has a average total synthesis error of around 100,000%. The large error for the single level population sets is a result of the number of solutions with low total area but incredibly high stiffness due in part to their compact design. The partitioning of the device into subsystems allowing each to focus on particular objectives, in this case stiffness error has improved performance; however a significant part is due to the addition of a constraint on how far each solution within the population set can deviate from the stiffness kx target of 2.45 N/m. This constraint is not present at the higher system level and the effect of this is seen through the almost sinusoidal oscillation of stiffness synthesis error as the system level population set is passed to subsystem two,

containing solutions with a higher stiffness error, and then quickly removed due to the local constraint on stiffness error.

Looking at the total synthesis error for both the system level population set of the MDO strategy and the single level population set in figure 6.68 it is clear that the MDO strategy is more successful in improving the synthesis error objectives than the single level strategy. Once again the disparity is a result of the single level strategy focusing heavily upon the total area objective, while the MDO strategy is structured to improve the overall synthesis error of the population. Each strategy is however able to find solutions which contain a low synthesis error, it is just the proportion of these solutions within the population and then how far the deviation is for some of the higher synthesis error solutions which make the difference between each strategy. The average synthesis error for the best solution found for both strategies is shown in figure 6.66 with similar performance throughout the first half of the design process, it is later that the MDO strategy is able to push on and find further improvement with an error of around 0.15% compared with 0.60% for the single level.

The structure of the design process, with a number of subsystems focusing on specific objectives and constraints can give rise to a local selection pressure that can drive the population sets to evolve solutions which have high local fitness, be they a design with a central mass that matches the target set out, or folded flexure topologies which provide the required stiffness outlined in the design problem. Ultimately the goal is to provide solutions which contain the elements that give rise to the associated high fitness, the central mass or folded flexure components, and retain them for the higher system level population set. In figure 6.68 it has been shown how the MDO strategy is able to reduce the synthesis % error for each of the mass and stiffness kx error objectives within each subsystem, and that the overall synthesis error for the system level population set is also decreased over the design process. In order to find a solution with a low total synthesis % error it is necessary for the evolved solutions from each of the subsystems to cooperate, so as to allow high fitness central mass components to be integrated with high fitness folded flexure components and vice versa.

Object based decomposition allows for decision variables, or genes which may have high linkage to be predetermined and passed on between parents and offspring. In this device level problem there is a clear linkage in function for the decision variables for the central mass and folded flexure resonator. The transfer of these genes from solutions in each subsystem is in essence a second crossover, with the hopeful outcome of combining decision variables associated with good mass error fitness to those with good stiffness kx error. Figure 6.69 shows two examples of the generational evolution of the best synthesis error % design found by the NSGAII MDO strategy. In both runs there are clear examples of the effect of allele migration between solutions from one subsystem to another leading to an improvement in the overall synthesis error of the new solution. Both run 4 and 5 examples also show the effect standard variation has on evolving the solutions topology as seen in run 5 and cycle 11 system population solution varying the anchor placement length, and run 4, cycle 11 to 21 in altering the folded flexure spring.

Interestingly in order to have an effective exchange of functional components it is important that their structure is not drastically changed in respect to the folded flexure spring or a large

variation in mass occurs in respect of the mass error objective. Decision variables anchor placement length, thickness and central mass length all have an effect on the stiffness of the folded flexure resonator. Combining high fitness central mass alleles with high fitness folded flexure alleles can improve the total synthesis error if the decision variables for central mass and thickness are similar as seen in run 5 at cycle 8 and run 4 at cycle 21. However there are instances where a little variation can be beneficial as seen in run 4, cycle 23, where the increased central mass length alters the folded flexure shape and improves the stiffness and total synthesis error.

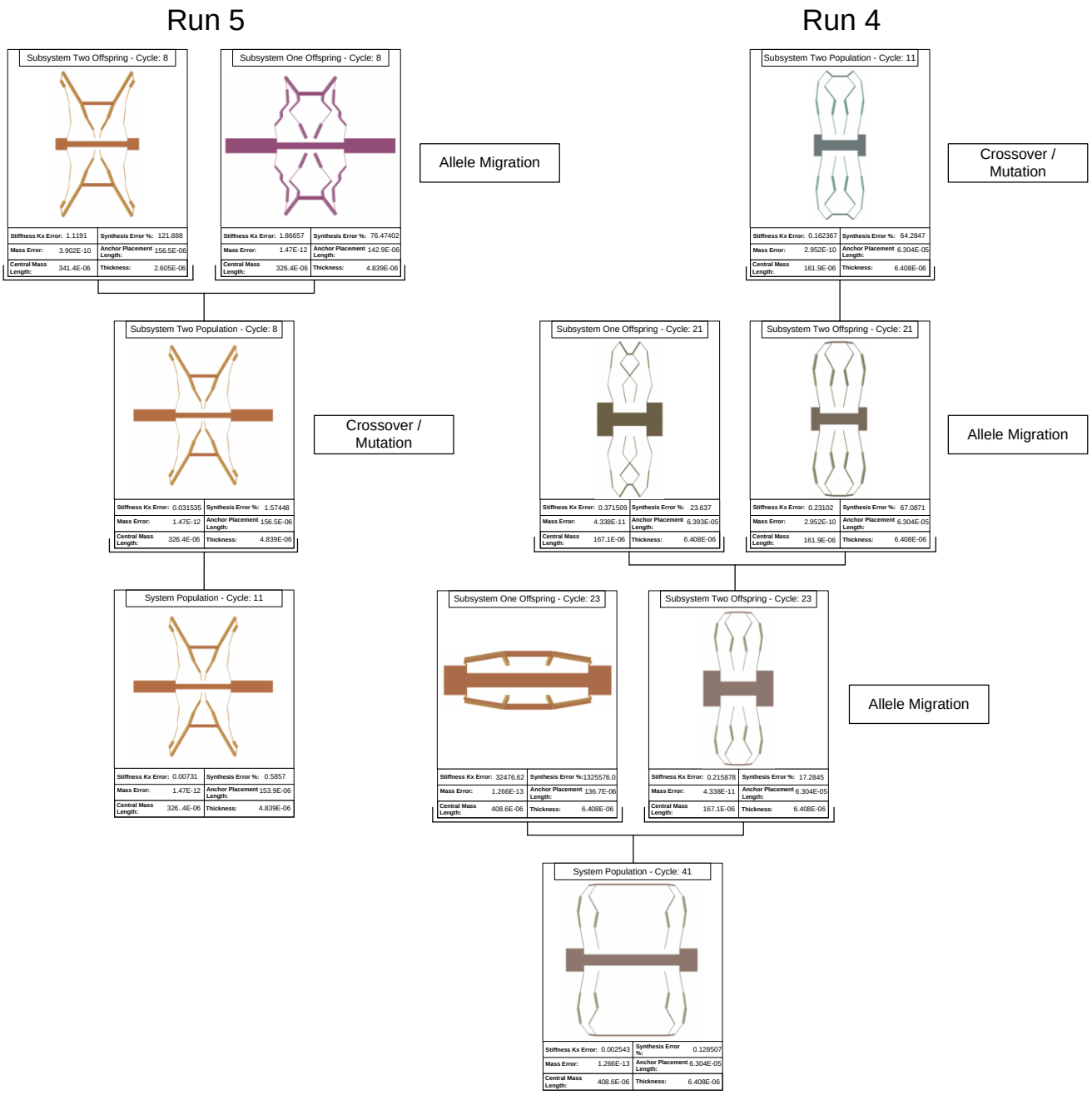


Figure 6.69 Genotype / Phenotype generational spanning trees for best synthesis error % solution for NSGAII MDO runs 4 and 5.

6.4.8 Device level comparison and analysis

The automated design synthesis and optimisation of 2D layout MEMS devices is one of the main challenges faced in the field of MEMS. The ability to simplify the process and open it up to application designers, break up complex devices into a number of functional units or simply help speed up the design cycle is of great benefit to MEMS design.

The design synthesis and optimisation of a MEMS folded flexure resonator device was undertaken using a variety of single and multi-level optimisation strategies. The performance of the multi-level strategies was once again superior to the current state of the art single level strategies for both NSGAII and SPEA2 algorithms. Table 6.41 lists the hypervolume performance values for all the strategies employed for the device level design problem for both NSGAII and SPEA2 algorithms.

Each strategy had its own flavour of optimisation with both benefits and drawbacks. The multi-level evaluation strategy allowed for more time to be spent exploring the design search space through a marked increase in the number of functional evaluations available to the optimizer. This allowed the best designs in terms of cost through a reduction in total area to be evolved, though due to the use of a restricted analytical folded flexure resonator model there are some elements of multi-level parameterization given the rigid topology employed. The multi-level parameterization strategy itself showed how a structured optimisation procedure which contains many levels of reduced parameterization and allows the flow of solutions form the more standard folded flexure resonator topologies to more adaptable and open ones could benefit design. Finally many MEMS devices can consist of many components or units which provide some level of function or behaviour. Here the folded flexure resonator can be broken down into two separate components, the central mass and the suspended folded flexure springs. This provided an opportunity to evaluate an object-based decomposition multidisciplinary optimisation strategy. The results were mixed in this case, with the ability to find many more designs that contained an error of less than 1% for each synthesis objective than all other strategies but also having poor hypervolume performance. Still the MDO process demonstrated its ability to evolve and optimise separate components and recombine them to produce optimal designs.

Table 6.41 Device Level Hypervolume Results for Single and Multi-level Strategies for Both NSGAII and SPEA2 Algorithms.

Device Level NSGAII				
Hypervolume	Single Level	Multi-Level Evaluation	Multi-Level Parameterization	Multidisciplinary Optimisation
S^U	1.29963645850762e-07	1.29985833605521e-07	1.29960502201653e-07	1.29553948363743e-07
S^M	1.2994629E-07	1.2997164E-07	1.29953601E-07	1.2898315E-07
S^L	1.29929021045027e-07	1.29942597914e-07	1.29945055166761e-07	1.28466481703699e-07
Device Level SPEA2				
Hypervolume	Single Level	Multi-Level Evaluation	Multi-Level Parameterization	Multidisciplinary Optimisation
S^U	1.29929836220047e-07	1.29932915126865e-07	1.29935645705392e-07	1.2970688951989e-07
S^M	1.2985024E-07	1.2958132E-07	1.29914260E-07	1.2902204E-07
S^L	1.2974065198587e-07	1.28742002912404e-07	1.29899070681417e-07	1.28572875228194e-07

* (S^U S^M S^L) [$1.3e^7$, $1e^{-9}$, $1e^{-5}$]

One of the most pressing features of the design optimisation of the folded flexure resonator undertaken by the SPEA2 algorithm across both single and multi-level strategies was its inability to explore and exploit regions of low synthesis error within the objective space. The lack of any solutions which exhibited an error of less than 1% for each of the synthesis objectives by the SPEA2 algorithm was caused in part due to the approach SPEA2 handles population sets which are fully non-dominated and therefore only differentiated by their SPEA2 distance metric. In contrast the NSGAII algorithm did not suffer with this particular problem and looking over the final population sets in figure 6.39 is able to produce more solutions which feature low synthesis error, especially when compared with those found by the SPEA2 approach. NSGAII employs a crowding distance metric and within this calculation each solution has its density to all of the nearest surrounding solutions for each objective calculated forming what is essentially a 'cuboid' whose size is used to calculate the distance metric value. An important feature to this is the normalization of the objective values for the population set so each objective is treated equally, something not applied to the SPEA2 distance metric.

As discussed in section 6.4.1 the stiffness kx error objective for solutions within the population set have a large range, often from $1e^{-2}$ to $1e^{5}$, and solutions which contain a low error in the region of $1e^{-2}$ will have neighbours which have a very small distance than those solutions that lie near $1e^{5}$ for this particular objective. Normalizing each of the objective values within the SPEA2 distance metric calculation may remove some of the bias towards solutions with a higher stiffness error.

In order to test this out the device single level experiments for SPEA2 were repeated, with one simple change to the distance metric calculation, to normalize all objective values. The results for the final Pareto set and solutions which were found to have an error of less than 1% for the two synthesis objectives are shown in table 6.42. Unfortunately in 3 out of the 5 experiments solutions which matched this criteria were found but later removed, a similar problem often occurred with the NSGAII algorithm, listed in brackets are the number of solutions found and later removed and their objective values are shaded. Only run 3 was able to retain a solution with the required low synthesis error at the end of the experiment. Still this is an improvement over the standard SPEA2 approach as it was never able to find a solution which contained an error of less than 1% in all of the single, multi-level evaluation and parameterization strategies.

This improvement in performance is shown in the hypervolume values obtained by the new normalized SPEA2 in table 6.43, providing superior performance in both the mean and boundary examples and outperforming the single level NSGAII in mean performance as well. The generational plot for the average hypervolume performance for both SPEA2 and normalized SPEA2 are shown in figure 6.70 showing a similar improvement in performance over the 5 runs. Also shown in figure 6.70 is the average synthesis error % for the best solution found over the design process for both SPEA2 and normalized SPEA2 algorithms. A clear improvement over the standard SPEA2 approach solutions of a lower synthesis error are able to be retained unlike the standard SPEA2 approach which converges to a solution on average and remains stuck there throughout the second half of the design process.

Table 6.42 Device Level Normalized Best Results Ranked by Total Area for Solutions with ≤ 1% Error

Device Level SPEA2 Normalized					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	Mass Error	Total Area
1	100	0 (2)	0.00412328	1.3973E-12	3.75802E-07
2	100	0	**NOT APPLICABLE**		
3	100	1	0.0064509	1.8523E-13	3.09684E-07
4	100	0 (1)	0.0087305	2.8696E-13	2.50066E-07
5	100	0 (1)	0.0017922	7.9626E-13	2.32581E-07

Table 6.43 Device Level Hypervolume Results for SPEA2 and SPEA2 Normalized

Device Level		
Hypervolume	SPEA2	SPEA2 Normalized
S^U	1.29929836220047e-07	1.29972800930411e-07
S^M	1.2985024e-07	1.2995829e-07
S^L	1.2974065198587e-07	1.299402207793e-07

* (S^U S^M S^L) [$1.3e^7$, $1e^{-9}$, $1e^{-5}$]

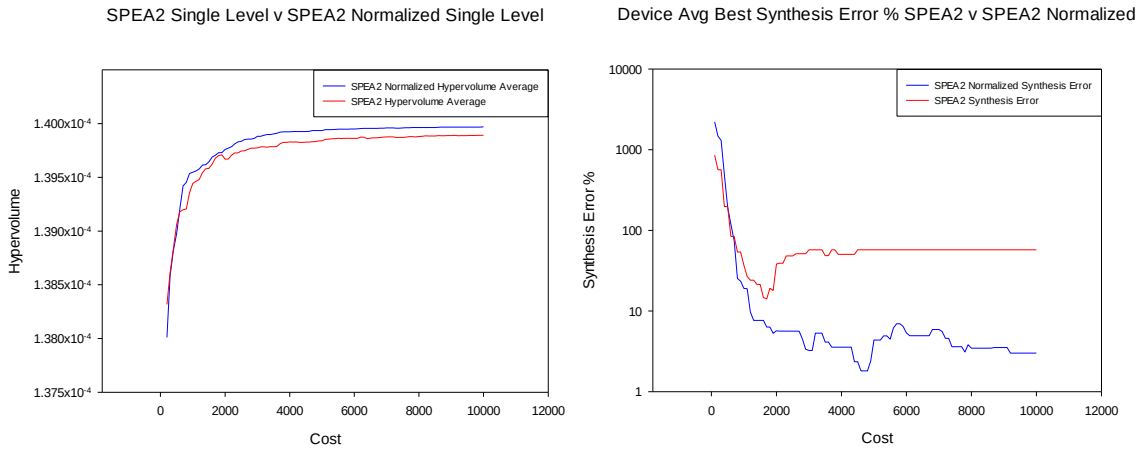


Figure 6.70 Device level average hypervolume results (left) for the 5 runs of the single level SPEA2 and SPEA2 normalized strategies and (right) average best synthesis error % for the single level SPEA2 and SPEA2 Normalized Strategies. * (S^U S^M S^L) [1.4e7, 1e-6, 1e-5].

6.5 Physical level design optimisation

The physical level occupies the most computational expensive and often most accurate level of modelling and analysis available to a MEMS designer [16]. Here three dimensional models are often employed using finite and / or boundary element analysis (FEA/BEA) over a number of multidisciplinary phenomena. The need for modelling and analysis at this particular level will only increase as more complex MEMS devices utilize or exhibit multiple disciplinary behaviour, often closely linked, and require tools such as COMSOL [91] to provide the necessary analysis needed to automatically synthesize the device in-silica.

Some of the most common planar MEMS devices are often constructed using a number of plate and spring components in order to provide a device which meets some functional requirement. Mechanical springs have been employed in a number of MEMS applications, from high

performance probing devices [292], MEMS transducers [293], microgyroscopes [294] and the coupling of resonators in 2D resonator arrays for MEMS filters [295] as employed within this thesis. Figure 6.71 outlines a number of common mechanical coupling springs, which include the simple single beam, folded flexure and serpentine spring topologies.

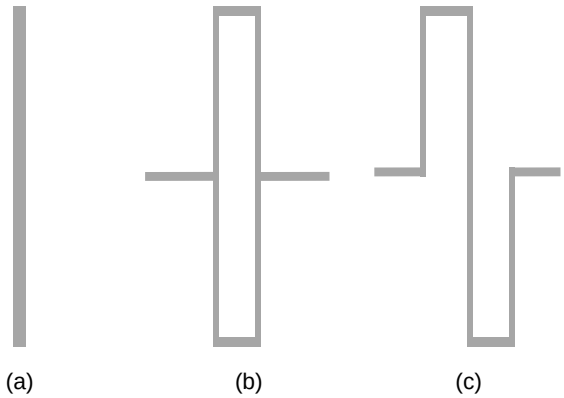


Figure 6.71 Coupling spring topologies – (a) single beam [131], (b) folded flexure [50], and (c) serpentine [296].

One requirement of a mechanical spring, particularly in coupled sensor devices, is the need to ensure no unwanted eigenmodes are introduced within the main modal frequencies of the structure [296]. The displacement of the resonator component along the axis of excitation is dependent upon the coupled mechanical springs and their functional behaviour, in particular the need for flexibility along the axis of excitation and rigidness in all others [296]. In terms of design synthesis the spring should also be able to be configured to a number of stiffness design targets dependent on the needs of the application and sensor functional behaviour. Mechanical springs such as the serpentine and folded flexure example can introduce spurious responses due to torsion and elongation of the spring during excitation leading to a reduction in sensor response [296][297]. This acted as motivation for creation of a novel ‘butterfly’ shaped mechanical spring for the application within a coupled sensor device in [296]. The new spring reduced unwanted eigenmode responses and had a more balanced displacement in comparison to the serpentine mechanical spring [296].

An important component with the bandpass filter device is the coupling spring used to couple individual resonators of the device. The design and optimisation of a MEMS component at the physical level focuses upon the coupling spring, and in particular the butterfly mechanical spring discussed previously. The butterfly spring has been modelled in both the NODAL analysis tool Sugar and the physical level FEA/BEA tool Comsol and is shown in figure 6.72. The model is uniform over both tools and is constructed as a number of connected beam elements which have been parameterized in preparation for design optimisation and is shown in figure 6.73.

The design parameters, objectives and constraints for the physical level design problem are outlined in table 6.44, with the target value for the stiffness kx error synthesis objective found in table 6.45.

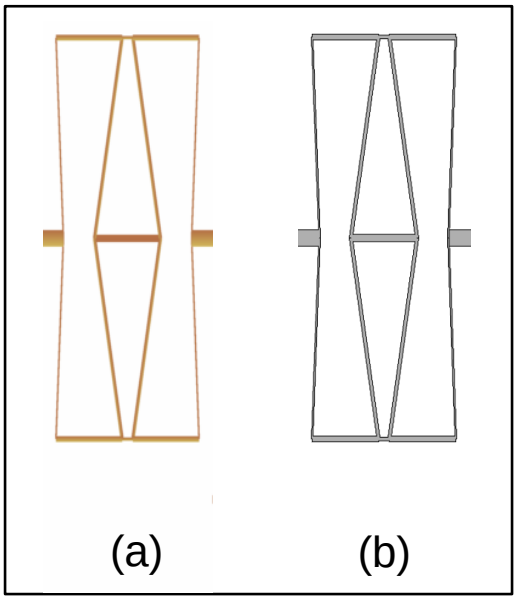


Figure 6.72 Butterfly coupling spring models for (a) Sugar and (b) COMSOL analysis tools

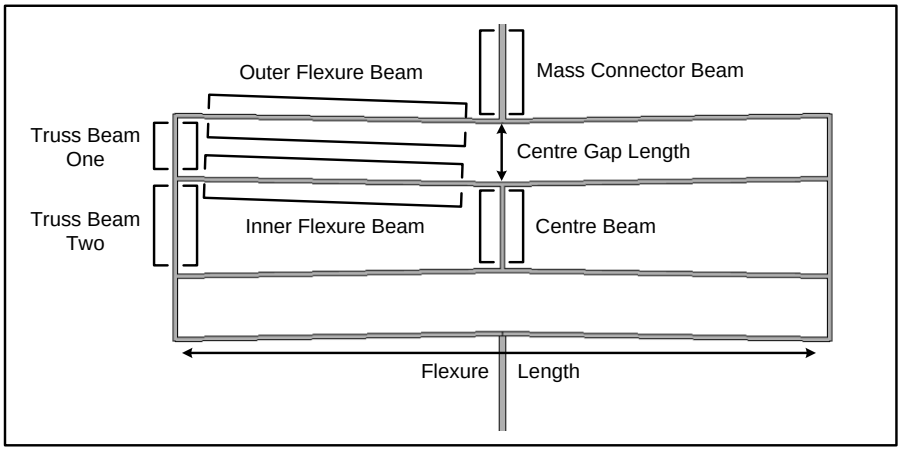


Figure 6.73 Parameterized butterfly coupling spring

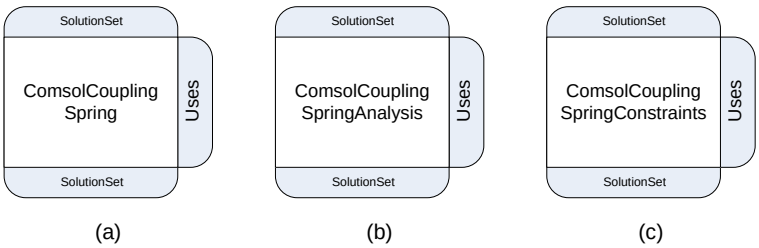


Figure 6.74 Physical level modules for comsol coupling spring problem information (a) comsol coupling spring analysis (b) and comsol coupling spring constraints (c).

Table 6.44 Physical Level Problem Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Truss Beam One Length (μm)	Real Valued	5	100
Truss Beam One Width (μm)	Real Valued	1	10
Truss Beam One Length (μm)	Real Valued	5	200
Truss Beam One Width (μm)	Real Valued	1	10
Mass Connector Length (μm)	Real Valued	2	100
Mass Connector Width (μm)	Real Valued	1	20
Centre Beam Length (μm)	Real Valued	20	100
Centre Beam Width (μm)	Real Valued	2	10
Outer Flexure Beam Width (μm)	Real Valued	1	10
Inner Flexure Beam Width (μm)	Real Valued	1	10
Flexure Length (μm)	Real Valued	50	400
Centre Gap Length (μm)	Real Valued	10	50
Thickness (μm)	Real Valued	1	5
Objectives		Constraints	
Coupling Spring Stiffness Kx Error	Minimize	Constraint 1 [Truss Beam Two Length – (Inner Flexure Beam Width * 2)]	Inequality ≥ 0
Von Mises Stress	Minimize	Constraint 2 [Centre Beam Length – ((Inner Flexure Beam Width * 2) + 3e-06)]	Inequality ≥ 0
		Constraint 3 [((Centre Beam Length / 2) + Centre Gap Length + Mass Connector Length) – ((Truss Beam Two Length / 2) + Truss Beam One Length + 10e-06)]	Inequality ≥ 0

Table 6.45 Physical Level Synthesis Objective Target Information

Synthesis Objective	Target Value
Coupling Spring Stiffness Kx Error	21.55 N/m

The objectives chosen firstly focus upon the requirement to synthesis a coupling spring component to match a target stiffness value that is a requirement for the coupled bandpass filter device. The second objective relates to attributes of the physical level modelling and analysis tools, with the goal of minimising the average von Mises stress levels of the coupling spring component. Minimising the von Mises stress could increase the life span of the component as its tolerance to fatigue through mechanical action is increased. This particular analysis is not found within system or device level tools and requires the use of computationally expensive FEA/BEA. It is important as a designer to be able to undertake analysis such as this; however it does have implications for the automation of design, particularly for population based optimisation algorithms which require a large number of functional analysis calls over the design process.

In order to undertake such analysis at the physical level, the modelling and analysis tool Comsol [91] was incorporated within the design framework. The new modules for the physical level are shown in figure 6.74 and include the Comsol problem information, analysis and constraint functions required to undertake the design optimisation of the coupling spring.

The model built within Comsol is derived from the topology and sizing first established within [296] and utilizing the same material and analysis properties established for the similar Comsol MEMS solid mechanics model example [91]. Table 6.4.1 contains some of the properties for the default coupling spring model.

Table 6.46 Comsol Butterfly Spring Model Attributes

Comsol Property	Attribute
Material	Poly-Si
Solid Mechanics	Linear Elastic Material Model
Force	Face Load 1 N/m
Mesh Quality	Coarse
Mesh Type	Free tetrahedral
Degrees of freedom	46719
Solver	Solid Mechanics
Simulation Time	6.0 Seconds

The automated synthesis of the coupling spring FEA model through the Comsol analysis tool and the evolutionary computation optimization routine requires a balance of accuracy and computational cost. A mesh quality of 'coarse' was chosen to allow for a reasonable level of accuracy at around 50,000 degrees of freedom, while keeping simulation time and memory resources to a minimum. The MEMS solid mechanics analysis is called and both the displacement values and average von Mises stress values are calculated as shown in figure 6.75. The displacement is taken from a point where the right mass connector beam meets the right fixed beam component. The coupling spring within the bandpass filter is normally connected to an input and output resonator which flanks it either side, however modelling the entire folded flexure resonator is not practically feasible. Instead a simple fixed beam spring as used in the original implementation is used to represent the resonator and the resistance their effective stiffness brings. In the bandpass filter a comb transducer applies a force upon the folded flexure resonator causing it to displace and in turn displace the coupled mechanical spring. To replicate this, a force of 1/Nm is applied along the face of the left fixed beam creating a displacement which can be used to measure the stiffness of the coupling spring. In addition the total von Mises stress throughout the whole coupling spring is calculated, with both left and right fixed beams ignored from this calculation as seen in the bounding box in figure 6.75b. The analysis of von Mises stress looks to calculate the three principle stresses (x,y, and z) of a load upon a structure in order to assess the yield strength of the material. If the von Mises stress exceeds the yield stress, then the material is considered to be at the failure condition.

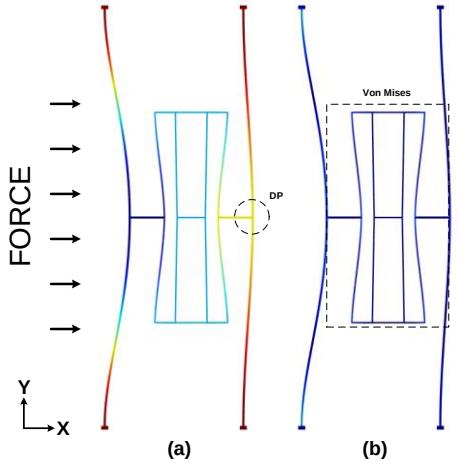


Figure 6.75 Comsol butterfly spring analysis (a) displacement X and (b) von Mises Stress

Over the life time of a MEMS device the continued function, in the case of the bandpass filter mechanical displacement, can lead to 'wear and tear' as a result of the stress placed upon it during activation. It is important to be able to design MEMS devices which are robust and durable enough to withstand frequent activation over the lifetime of the device without failing through mechanical stress fracture. The addition of physical level tools which allow for this kind of analysis provide designers the ability to factor in the effect the sizing and topology of the coupling spring has on mechanical stress levels, and therefore optimize the design in order to minimize the effect.

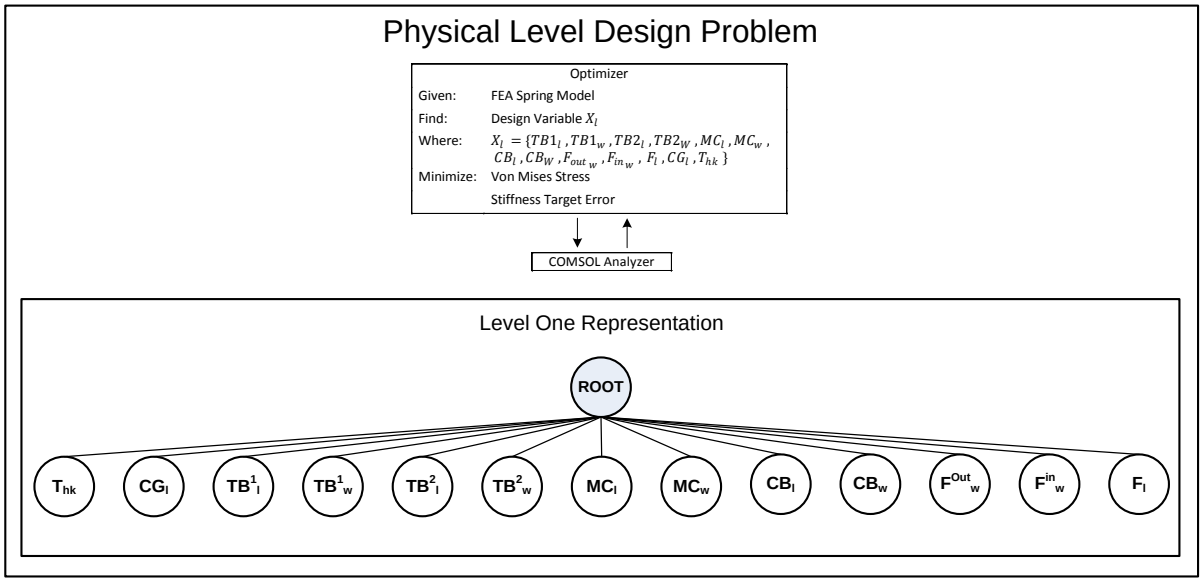


Figure 6.76 Physical Level design template, with overview of problem and default representation.

The overall physical level design process is shown in figure 6.76 and contains the default representation used by the optimization algorithms. The NSGAII and SPEA2 algorithm parameters remain the same as in the previous single level examples with the only exceptions being the removal of structural crossover operators and changes to the population set sizes and generations as shown in table 6.47. The change in both population size and the number of generations reflect the extra computational cost of analysis from the physical level tool Comsol and bring about a similar level of time required to complete the design process with those at the system and device level previously.

Table 6.47 Physical Level Algorithmic Parameter Changes

Algorithm Parameter	Default Value
Population Size	50
Offspring Size	50
Selection Size	50
Replacement Size	100
Generations	20

6.5.1 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.77 and fully in appendix C.3. A list of the best results ranked by von Mises stress objective values and constrained to having a stiffness error of less than 1% from the target for each run are shown in table 6.48 and their phenotypes shown in figure 6.78 and figure 6.79. Finally the hypervolume values for both algorithms are shown in table 6.49, with the best results shaded.

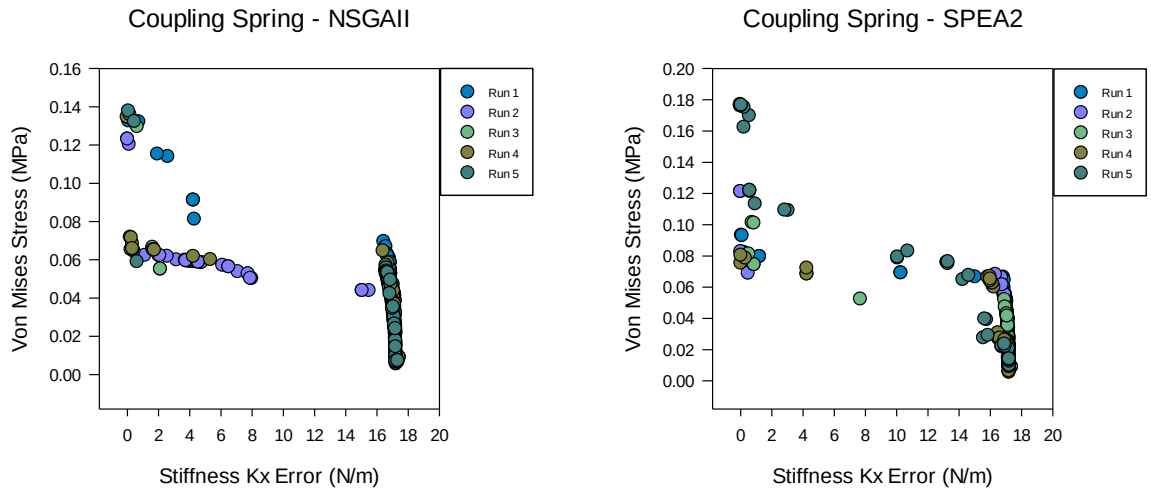


Figure 6.77 Physical level run 1 – 5 final population sets for (left) NSGAII and (right) SPEA2

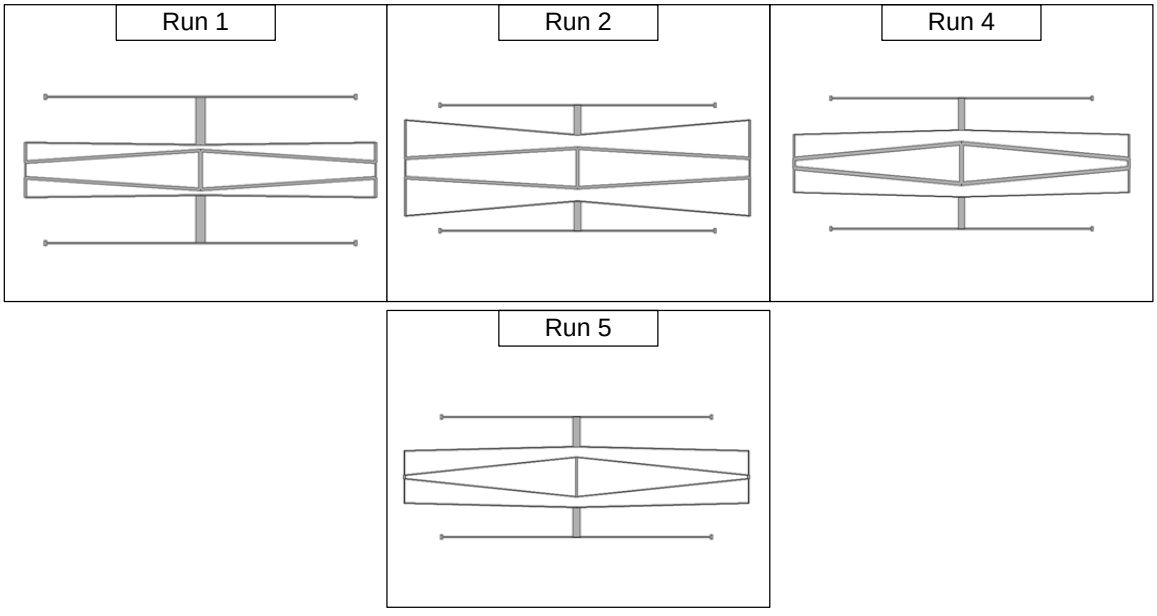


Figure 6.78 Physical level runs 1, 2, 4 and 5 coupling spring best result ranked by stiffness kx error for NSGAII

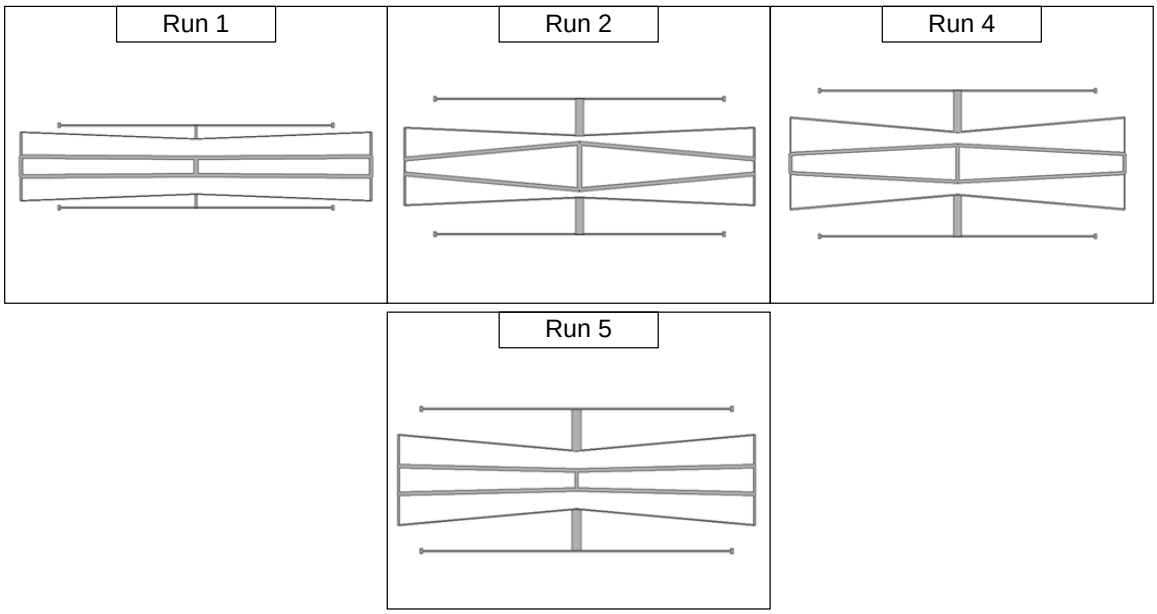


Figure 6.79 Physical level runs 1, 2, 4 and 5 coupling spring best result ranked by stiffness kx error for SPEA2

Table 6.48 Physical Level Best Results Ranked by von Mises Stress for Solutions with ≤ 1% Error

Physical Level NSGAII				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	von Mises Stress (MPa)
1	50	1	0.1098876	0.13234669
2	46	3	0.1398810	0.12005957
3	50	0	-	-
4	50	4	0.2134267	0.07155961
5	50	1	0.0963080	0.13758687
Physical Level SPEA2				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	von Mises Stress (MPa)
1	50	2	0.1258840	0.09274012
2	50	2	0.0360614	0.08248243
3	50	0	-	-
4	30	2	0.0403522	0.08013869
5	20	1	0.0013145	0.17578129

Table 6.49 Physical Level Hypervolume Results for NSGAII and SPEA2

Physical Level		
Hypervolume	NSGAII	SPEA2
S^U	19.009083302527	18.780618043083
S^M	18.6830137	18.5908500
S^L	18.3231287492148	18.3778343324101

* (S^U S^M S^L) [20, 1]

The design optimisation of a novel 'butterfly' coupling spring formed the basis of evaluation for the physical level modelling and analysis of MEMS and how they can be incorporated into an automated design process. The results for both NSGAII and SPEA2 algorithms include solutions which were able to have a target error for the stiffness kx objective of less than 1% with the best result ranked by the von Mises stress objective coming from run 4 of the NSGAII strategy. The

final population sets for both algorithms are very similar, with a clear Pareto front characteristic of solutions forming along an stiffness kx error of 17 N/m for low von Mises stress levels before solutions at higher stress levels are able to quickly converge to stiffness kx error values near 10% or less of the design target.

The phenotypes for the best solutions found for both the NSGAII and SPEA2 algorithms as seen in figures 6.78 and 6.79 show similar characteristics with a large design consisting of a long flexure length with a thicker and more rigid inner flexure beam core and a slimmer outer flexure beam shape. Figure 6.80 tracks a number of phenotypes over the objective space for run 2 of the NSGAII strategy. Solutions which lie closer to the target stiffness kx value contain this particular shape and topology while solutions which contain a much higher level of stiffness kx error often mimic the characteristics seen at the bottom of figure 6.80, with a reduced size and increased flexure beam thickness and rigidity. This particular phenotype being very rigid and stiff leads to a much larger displacement at the point of analysis and a much larger deviation from the target stiffness outlined in table 6.45, the benefit of this design is that there is much less deformation in the spring structure and therefore a lower von Mises stress value.

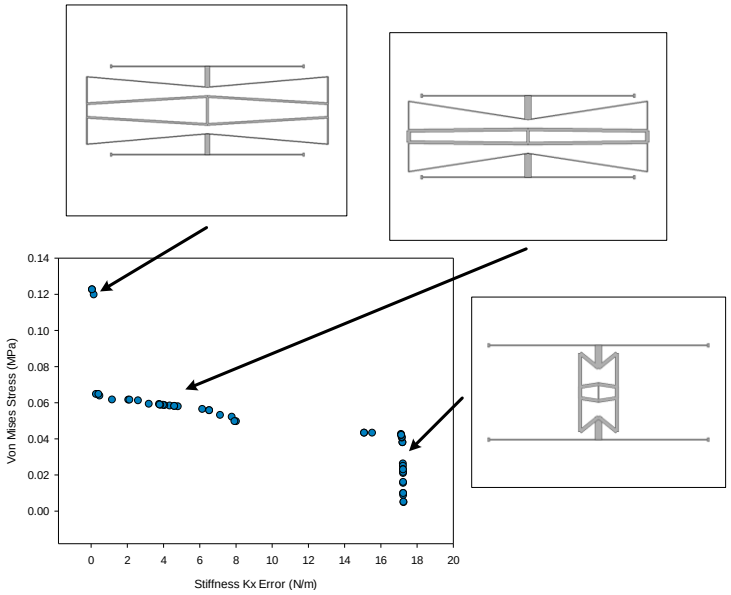


Figure 6.80 Physical level coupling spring objective space phenotypes NSGAII run 2

The performances of both algorithms are reasonably similar, in both hypervolume performance, the number of solutions within 1% error of the design target and their objective values. Figure 6.81 substantiates this assessment of performance with both algorithms converging to similar hypervolume performance values, though with NSGAII at a slightly faster pace early on in the design process.

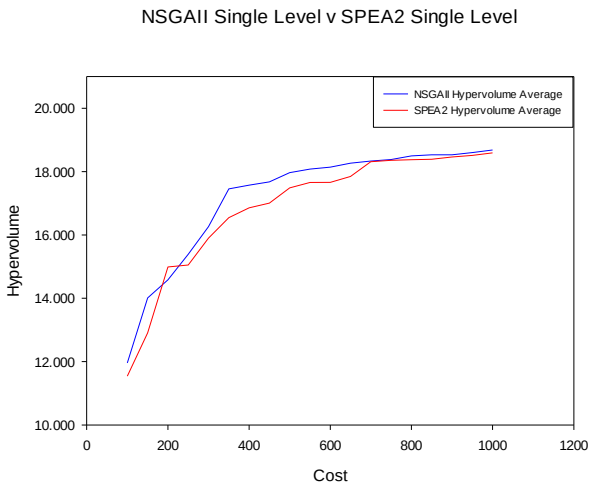


Figure 6.81 Physical level average hypervolume results for the 5 runs of the single level NSGAII and SPEA2 Strategies * ($S^U S^M S^L$) [20, 1]

6.5.2 Multi-level evaluation

The application of the finite element method through solid modelling and analysis tools like Comsol often requires large computational resources, particularly in simulation time and memory allocation. A large factor of this cost comes from the meshing of the model itself and the time and resources required to simulate any analysis. The chosen meshing type, in this instance tetrahedral, and the quality of the mesh, affect the number of nodes and degrees of freedom present within the model for simulation. Varying either of these parameters can lead to a change in the accuracy of simulation, with the lowest level of meshing quality giving lower degrees of freedom and simulation accuracy. However the cost of simulation also decreases with lower quality meshing taking less time and resources to undertake analysis comparatively with high quality meshes.

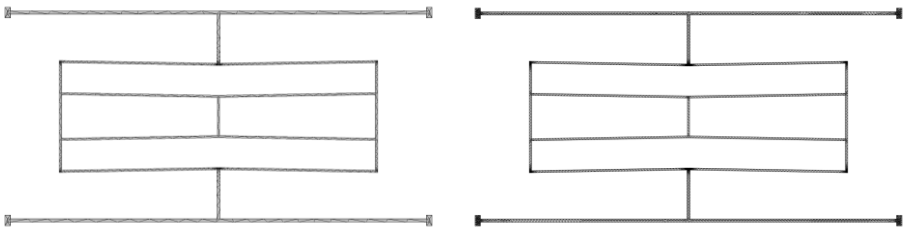


Figure 6.82 Comsol coupling spring meshing for (left) ‘extremely course’ and (right) ‘course’ parameters

As a result it is possible to model and analyse a MEMS device at a lower computational cost and reduced accuracy through the application of a lower quality meshing of the model as seen in figure 6.82, or increase the computational cost by modelling at a higher meshing quality with the aim of achieving more accurate analysis. It is however not as simple and clear cut to state that applying one level of meshing detail or another will give rise to an increase or decrease in

accuracy as it is more of an art than a science. However the quality of meshing, and in particular the number of nodes and their distribution play the most important part in the accuracy of analysis and for the purpose of this experiment provide a suitable representation of the required multiple levels of evaluation. The evaluation cost for each analysis call is given in table 6.50.

Two migrator modules are utilized in this particular design strategy to allow individuals to move between neighbouring levels. The values for migration percentage along with the cycle count when migration is invoked are shown in table 6.51. The low level pathway is called sequentially throughout the whole design process, while the higher level is called every two cycles to give a rough 1:1 ratio in terms of functional evaluation cost over the whole design process.

Table 6.50 Evaluation Cost for 'Extremely Coarse' and 'Coarse' Meshed Coupling Spring Model

Extremely Coarse Analysis	Coarse Analysis
Mean Analysis Time (20 Calls)	Mean Analysis Time (20 Calls)
3.0	6.0
Ratio	
1 :2	

Table 6.51 Migrator Module Parameters for Multi-Level Evaluation

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	20	4
Level 2	Level 1	20	4

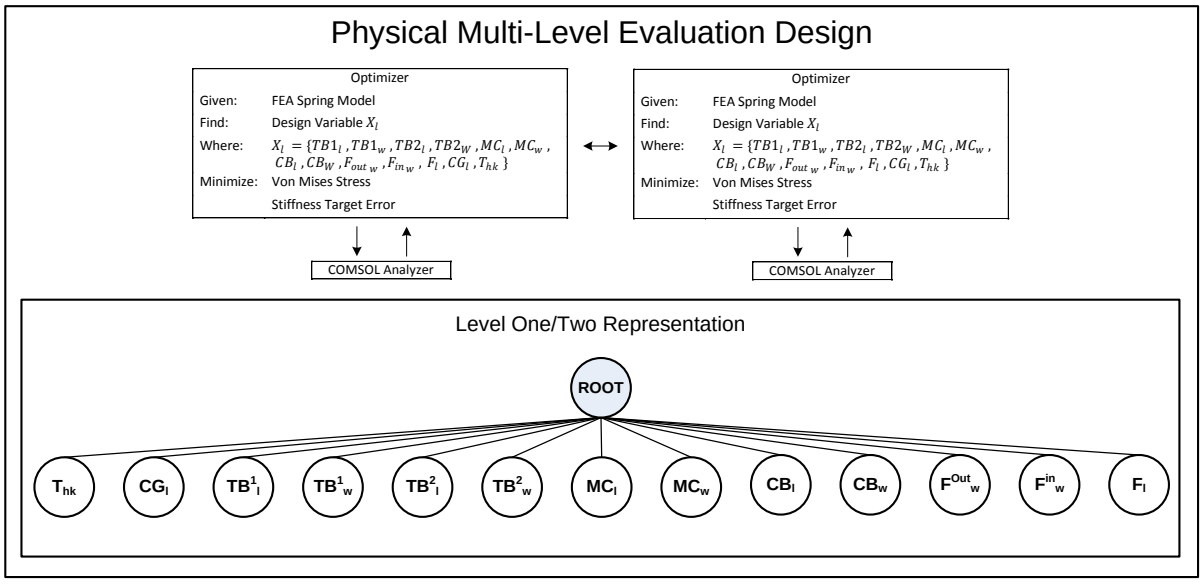


Figure 6.83 Physical multi-level evaluation design template, with overview of problem and default representation.

6.5.3 Numerical Results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.84 and fully in appendix C.3. A list of the best results ranked by von Mises stress objective values and constrained to having a stiffness error of less than 1% from the target for each run are shown in table 6.52 and their phenotypes shown in figure 6.85 and figure 6.86. Finally the hypervolume values for both algorithms are shown in table 6.53, with the best results shaded.

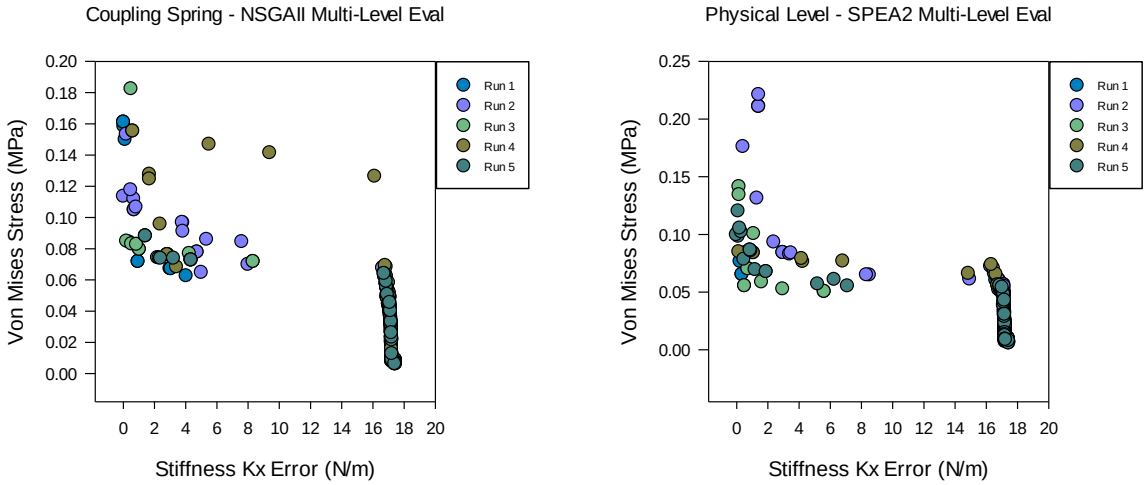


Figure 6.84 Physical level run 1 – 5 final population sets for (left) NSGAII multi-level evaluation and (right) SPEA2 multi-level evaluation

Table 6.52 Physical Multi-level Evaluation Best Results Ranked by von Mises Stress for Solutions with ≤ 1% Error

Physical Multi-Level Evaluation NSGAII				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	von Mises Stress (MPa)
1	50	4	0.043855	0.160846
2	37	1	0.03082727	0.1132739
3	40	0	-	-
4	40	0	-	-
5	40	0	-	-
Physical Multi-Level Evaluation SPEA2				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	von Mises Stress (MPa)
1	21	0	-	-
2	26	1	0.0449232	0.2157536
3	23	4	0.1414424	0.1021242
4	36	1	0.1622766	0.0845189
5	28	3	0.1281946	0.0979628

Table 6.53 Physical Multi-Level Evaluation Hypervolume Results for NSGAII and SPEA2

Physical Multi-Level Evaluation		
Hypervolume	NSGAII	SPEA2
S^U	18.7052154228448	18.94877986031
S^M	18.2432254	18.6494235
S^L	17.3919364330269	18.193898408987

* (S^U S^M S^L) [20, 1]

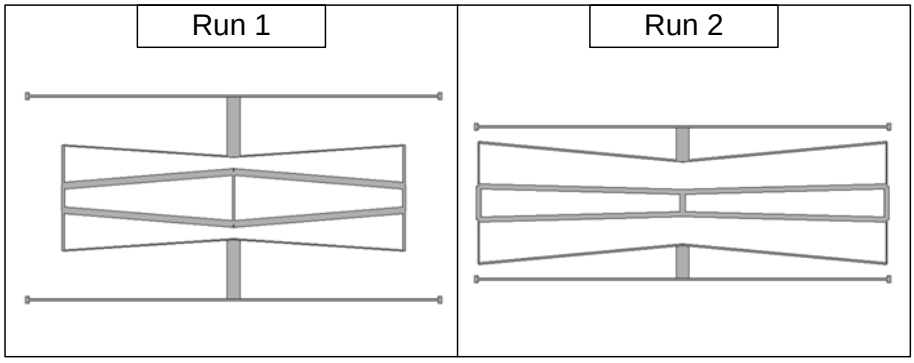


Figure 6.85 Physical level run 1 – 2 coupling spring best result ranked by stiffness kx error for NSGAII multi-level evaluation

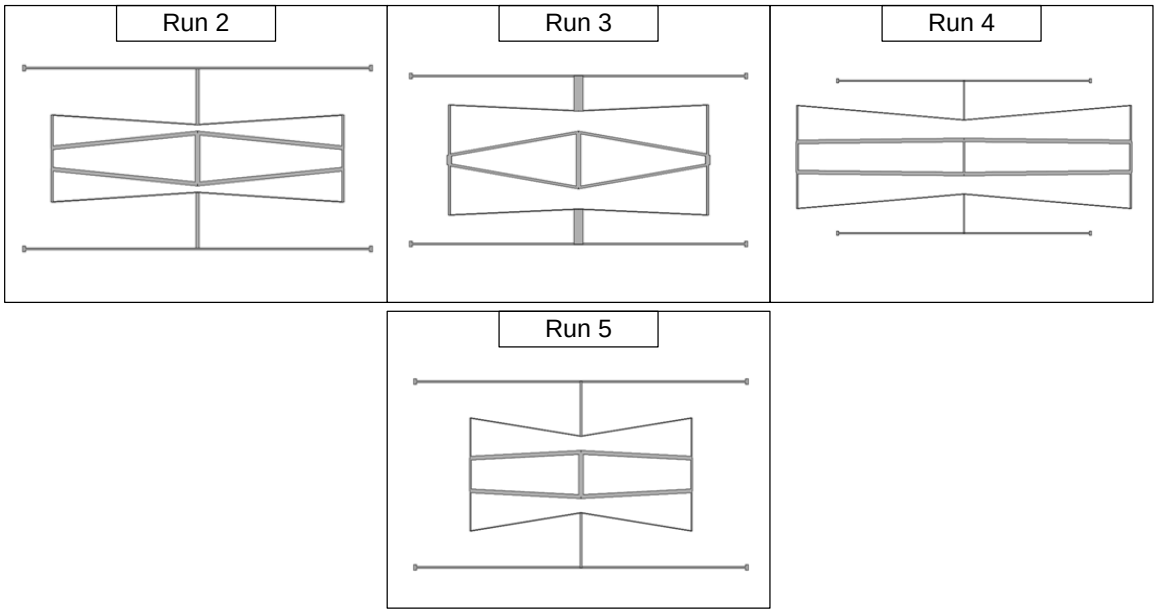


Figure 6.86 Physical level run 2 – 5 coupling spring best result ranked by stiffness kx error for SPEA2 multi-level evaluation

MEMS design optimisation at the physical level requires the efficient use of often limited computational resources throughout the design process. When the numbers of functional evaluations are low as a result of the computational cost, often in time, for a single evaluation it is important to increase the number when possible. A strategy of increasing the number of functional evaluations over the design process through reducing the cost of the individual analysis of each solution through two different meshing methods was undertaken. The hypervolume values of both the NSGAII and SPEA2 algorithms shows similar performance to their single level counterparts with hypervolume values either slightly ahead in the case of the multi-level SPEA2 approach or slightly below with the NSGAII method. This is mimicked with the generational hypervalue performance seen in figure 6.87 with only a reduction in the early part of the design process for each algorithm.

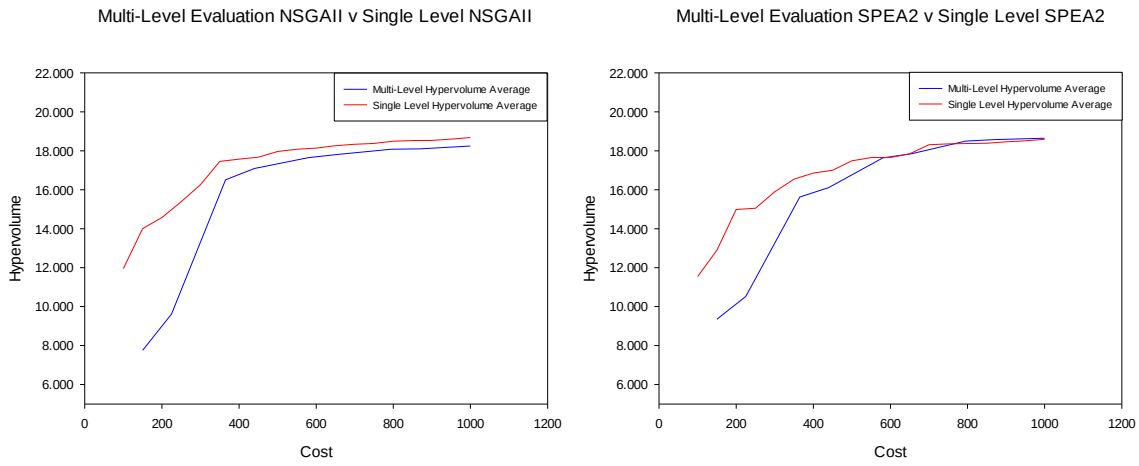


Figure 6.87 Physical level average hypervolume results for the 5 runs of the multi-level parameterization and single level NSGAII and SPEA2 strategies * ($s^U s^M s^L$) [20, 1]

The addition of multiple populations means convergence of the populations to areas of high hypervolume performance takes longer in relation to the overall cost than that of the single level method unless migration between the two populations is strong. Figure 6.88 shows the average migration success for both NSGAII and SPEA2 algorithms, with more time spent on evolving the low cost population it is not surprising to see more solutions successfully migrating to the higher level.

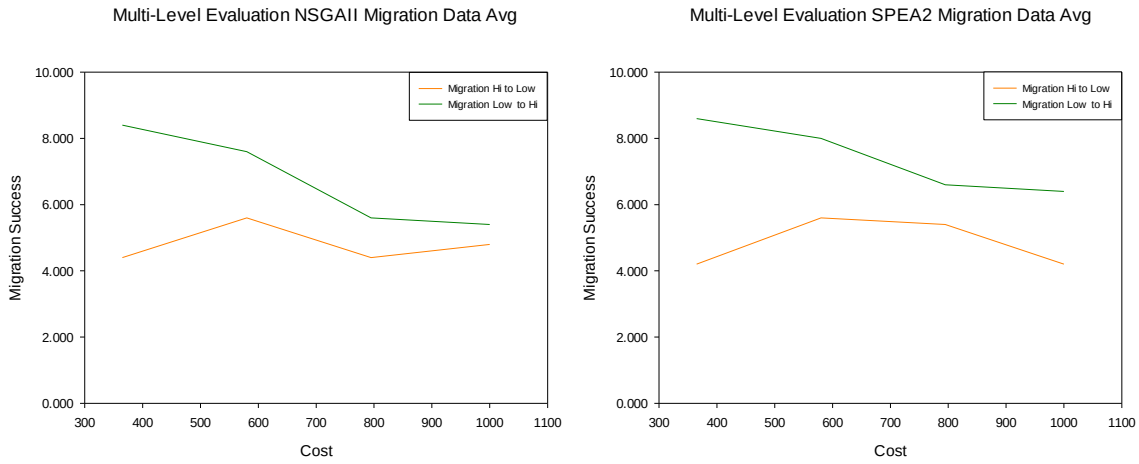


Figure 6.88 Generational average migration successes for multi-level evaluation NSGAII and SPEA2 strategies.

Given the low number of functional evaluations used throughout the design process even with the addition gained through the multi-level evaluation strategy the partition into two separate populations has probably outweighed any benefit the strategy can bring using this particular level of modelling and analysis. Overall the strategy gave around 500 additional functional evaluations or about 250 for each level with around 80 of these evaluations wasted on migrating solutions.

6.5.4 Multi-level parameterization

The parameterized coupling spring consists of thirteen individual decision variables which are used to control the sizing of the component. The larger the numbers of decision variables open to the optimization algorithm to evolve, the greater the degree of creativity and exploration it can undertake during the design process. However at the physical level the increased computational cost of simulation reduces the number of functional evaluations which can be undertaken over the design process in a practical time. Increased exploration is useful when the numbers of functional evaluations are large and convergence of the population through exploitation of good designs can occur over a longer period. The physical level design problem has a small number of functional evaluations comparatively to other design problems solved previously. Therefore the optimizer has to balance both exploration and exploitation within these limitations, too strong a selection pressure and the population will converge and be unable to find solutions which lie on the true Pareto front, too little a selection pressure and good designs will be underutilised slowing the design process down.

It may be possible to undertake both paths, by utilising multiple levels of parameterization, there can be levels where exploration is limited and levels where full exploration of the design search space can be undertaken. The multi-level parameterization strategy for physical level design focuses upon this approach by breaking the design process into two such levels.

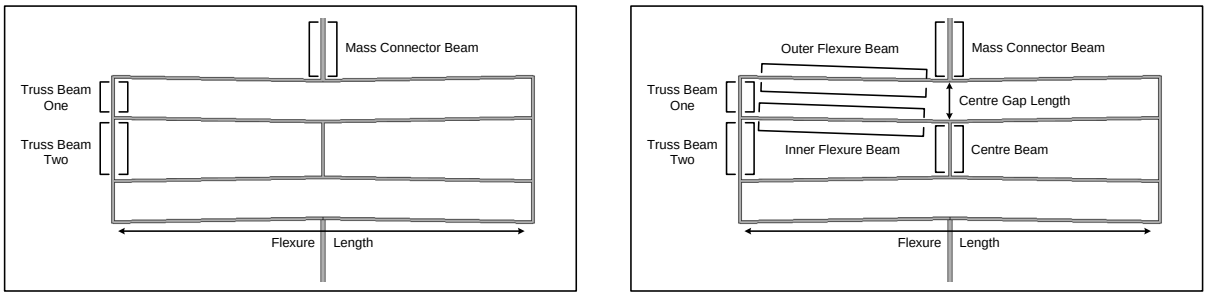


Figure 6.89 Physical multi-level parameterization representations for (left) level one and (right) level two

Each of these levels runs the same physical level design problem, however under different parameterized representations as seen in figure 6.89. Here level one consists of a coupling spring model with only six of the thirteen design variables open for evolution by the optimization algorithms, while level two contains the fully parameterized version.

Two migrator modules are utilized in this particular design strategy to allow individuals to move between neighbouring levels. The values for migration percentage along with the cycle count when migration is invoked are shown in table 6.54. The low level and high level pathways are called sequentially throughout the whole design process.

Table 6.54 Migrator Module Parameters for Multi-Level Parameterization

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	20	2
Level 2	Level 1	20	2

The multi-level parameterization strategy for the physical level design problem is outlined in figure 6.90 and contains the parameterized representations and node markers for each level. The algorithmic parameters for both NSGAII and SPEA2 remain unchanged with the whole design process still using 1000 functional evaluations.

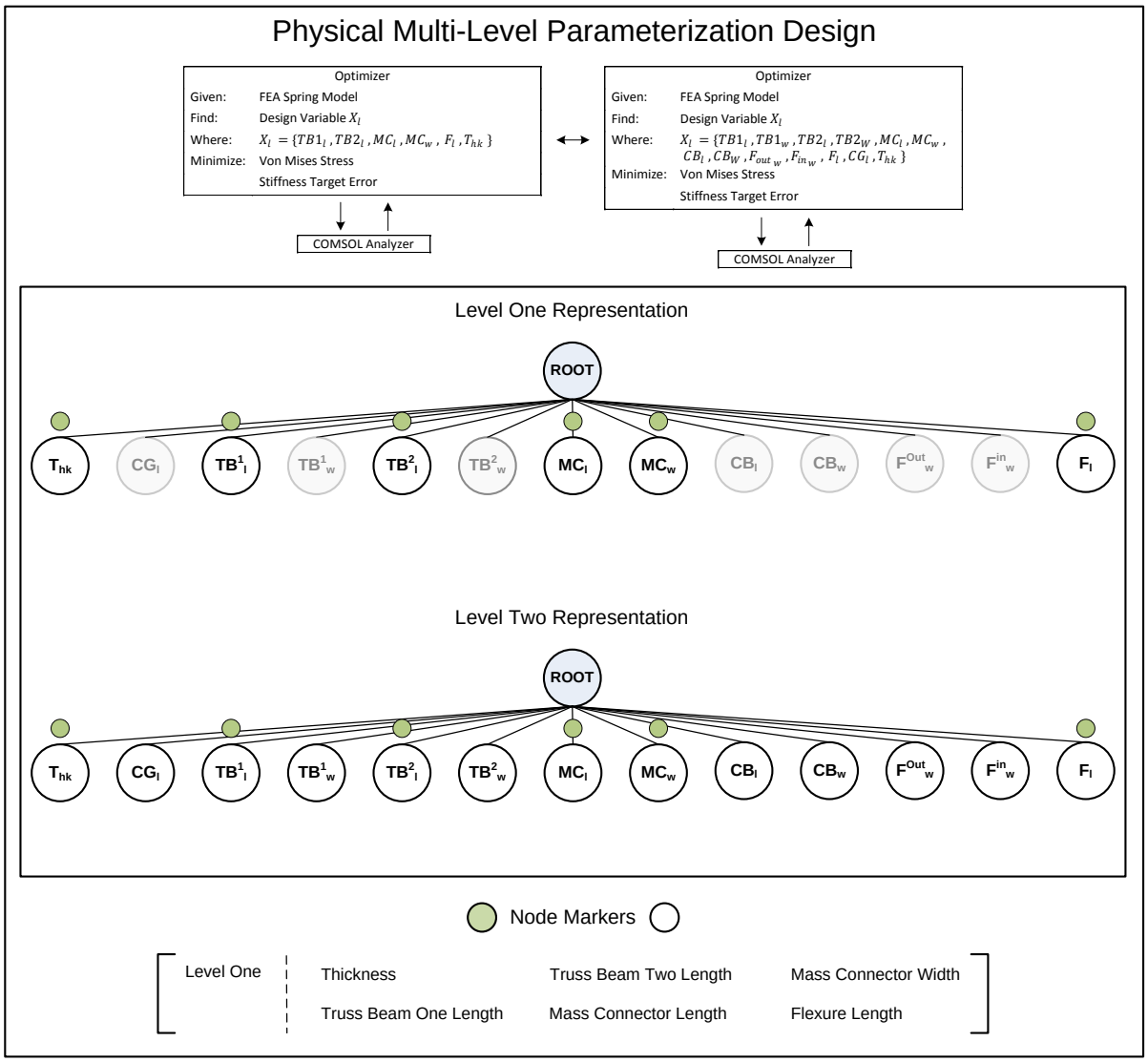


Figure 6.90 Physical multi-level parameterization design template, with overview of problem, default representations, and node markers.

6.5.5 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.91 and fully in appendix C.3. A list of the best results ranked by von Mises stress objective values and constrained to having a stiffness error of less than 1% from the target for each run are shown in table 6.55 and their phenotypes shown in figure 6.92 and figure 6.93. Finally the hypervolume values for both algorithms are shown in table 6.56, with the best results shaded.

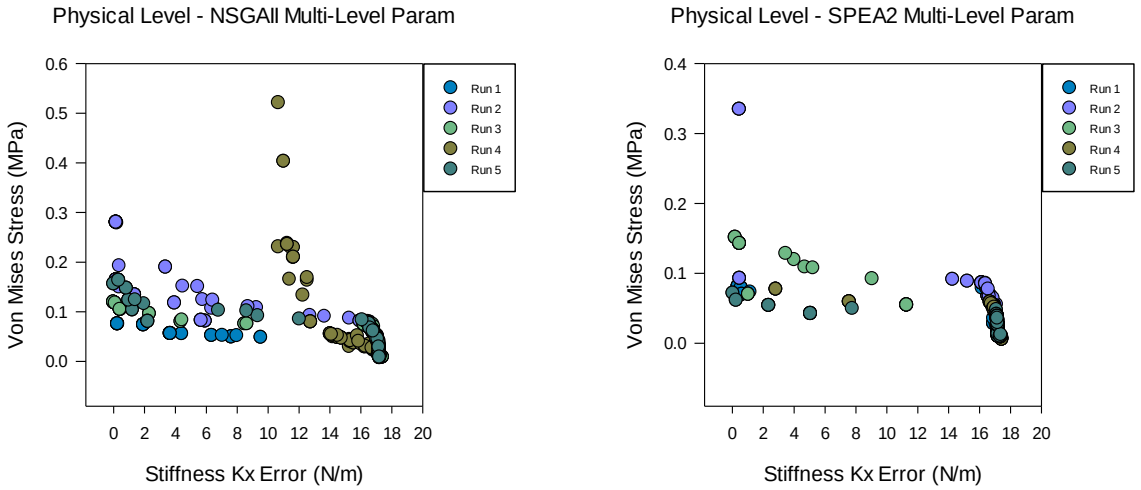


Figure 6.91 Physical level run 1 – 5 final population sets for (left) NSGAII multi-level parameterization and (right) SPEA2 multi-level parameterization

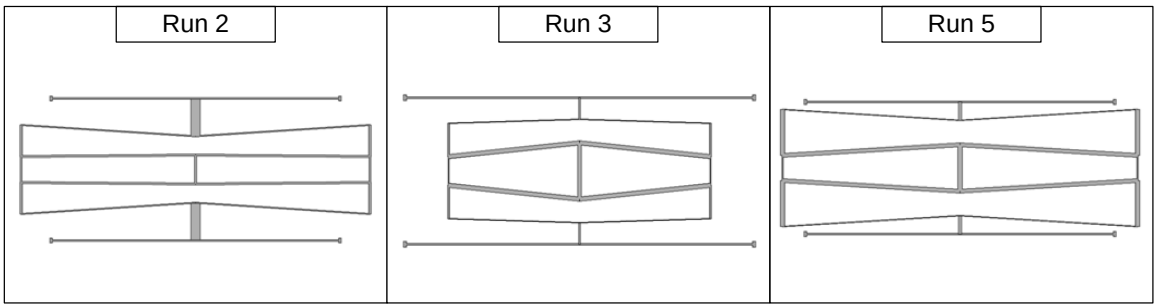


Figure 6.92 Physical level runs 2, 3 and 5 coupling spring best result ranked by stiffness kx error for NSGAII multi-level parameterization

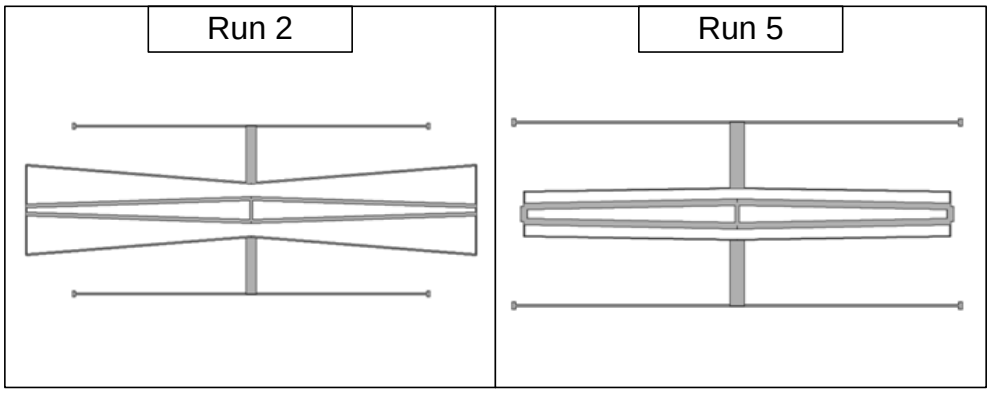


Figure 6.93 Physical level runs 2 and 5 coupling spring best result ranked by stiffness kx error for SPEA2 multi-level parameterization

Table 6.55 Physical Multi-Level Parameterization Best Results Ranked by von Mises Stress for Solutions with ≤ 1% Error

Physical Multi-Level Parameterization NSGAII					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Level	Stiffness Kx Error	von Mises Stress (MPa)
1	78	0	-	-	-
2	85	17	Hi	0.19715358	0.16328055
3	82	13	Hi	0.12376288	0.11681374
4	63	0	-	-	-
5	96	3	Hi	0.03491945	0.15567218
Physical Multi-Level Parameterization SPEA2					
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Level	Stiffness Kx Error	von Mises Stress (MPa)
1	32	0	-	-	-
2	74	1	Low	0.16283493	0.07926727
3	71	0	-	-	-
4	81	0	-	-	-
5	66	2	Hi	0.05643231	0.07098663

Table 6.56 Physical Multi-Level Parameterization Hypervolume Results for NSGAII and SPEA2

Physical Multi-Level Parameterization		
Hypervolume	NSGAII	SPEA2
S^U	18.7754454455268	19.1241410324346
S^M	16.53080026	18.07053509
S^L	8.70650139026446	16.2150877162596

* (S^U S^M S^L) [20, 1]

The design optimisation of a MEMS coupling spring using a multi-level parameterization strategy was the focus of this section with the results showing similar but lower performance with the ability of both algorithms to evolve final population sets which show similar characteristics to their single level counterparts. However the strategy lead to a lower convergence with neither level one or two population sets converging to a fully Pareto set and also having lower hypervolume performance in comparison with the single level itself. The addition of multiple populations did allow for some runs to increase the number of final solutions which contained an synthesis error of less than 1% but over the ten runs the robustness has dropped off with only 5 out of ten finding a low synthesis error solution.

The generational hypervolume results are shown in figure 6.94 for both algorithms against their single level methods with SPEA2 multi-level parameterization performing the better of the two multi level algorithms. The performance of level one and level two over the course of the design process are similar with a slight dip for the simplified representation level one halfway through the design process. The high level of migration between the two levels keeps each population set updated with the best solutions and the different levels of parameterization can lead to stepwise improvements over the design process as seen in figure 6.95. Here a single solution is evolved and varied over the design process through both levels of parameterization

However given the low number of functional evaluations available and the splitting of the search between two separate levels the overall gain from such a multi-level strategy is limited it would seem for a design problem of this kind.

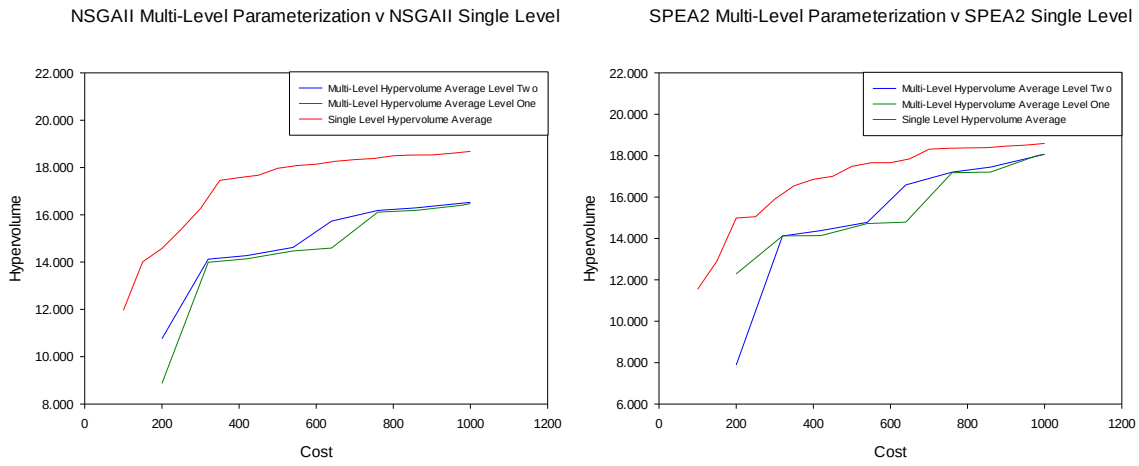


Figure 6.94 Physical level average hypervolume results for the 5 runs of the multi-level parameterization and single level NSGAII and SPEA2 strategies * (s^U s^M s^L) [20, 1]

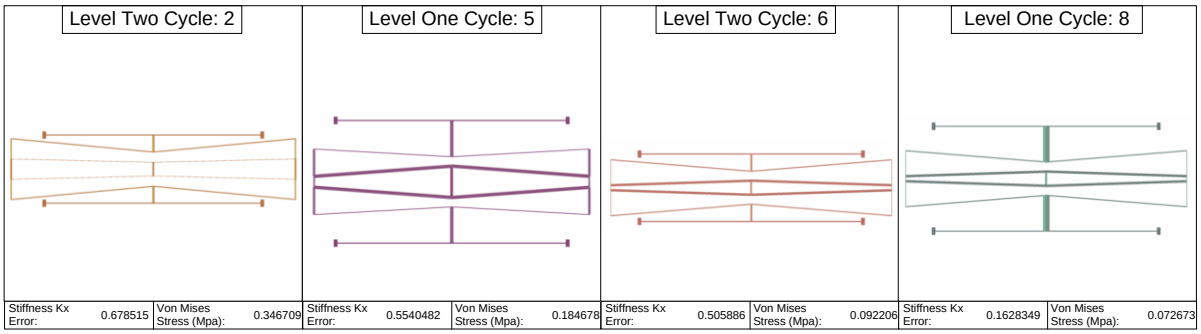


Figure 6.95 Generational phenotypes for the best solution found by the SPEA2 multi-level parameterization strategy run 2.

6.5.6 Multidisciplinary optimisation

The physical level of MEMS design optimisation often includes the largest number of disciplinary analysis associated with a particular device given the access to structural and mechanical analysis through FEA or the inclusion of more complex magnetic or fluidic analyses such physical level tools can provide.

The physical level design problem consists of two disciplinary analyses, structural for the calculation of the coupling spring's stiffness, and stress analysis for calculation of the average von Mises stress levels throughout the component. The construction of the spring and in particular its topology, provide certain design variables which can play a more direct role on the outcome for both the stiffness and stress characteristics of the coupling spring [296]. The object based decomposition of the spring into sections which reflect this influence, and the addition of new objectives to focus search upon these characteristics could provide a benefit to the overall design process. The physical level coupling spring design problem consists of thirteen decision variables which affect the shape and sizing of the component. An object based decomposition has deconstructed the spring into two separate sections as shown in figure 6.96 with a focus upon the outer flexure characteristics in subsystem one and the inner central core in subsystem two. The flexure length, width, thickness and truss beam one variables control how the outer

section of the coupling spring responds to an applied force with long, thin flexures providing lower stiffness and as a result increased displacement. The opposite is true with thick, short flexure beams providing characteristics which give a reduced displacement through an increased stiffness. Subsystem two contains variables for the central core of the coupling spring, a key component in the internal function of the spring in maintaining displacement along the axis of excitation and reducing the possibility of torsion and elongation along the other axes [296]. Subsystem two also controls the variables for the mass connector whose sizing can influence local stress value at the point of connection. The system level contains decision variables which can influence both subsystems, in particular through constraints and intersection of various flexures. Placing the centre gap length and central beam variables at the system level means there variation will be reduced but as a positive also reduce the chance of solutions failing constraints when gene swapping occurs.

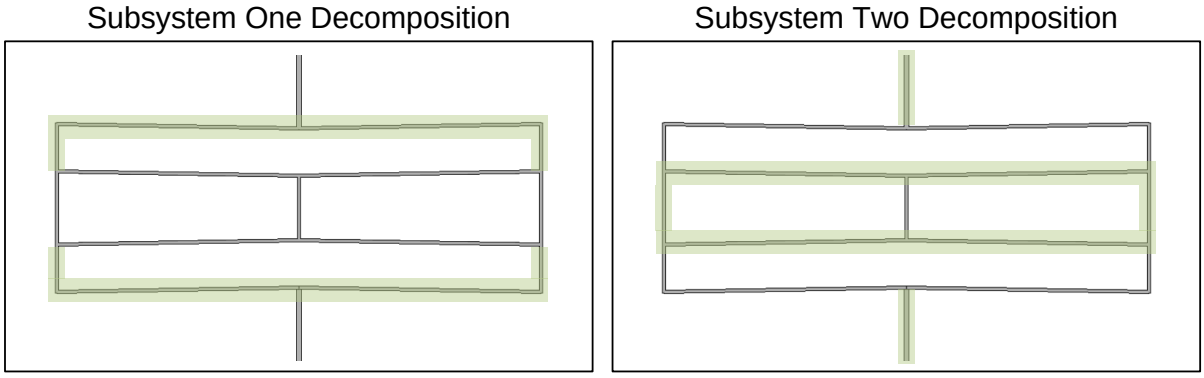


Figure 6.96 Physical multidisciplinary optimisation object decomposition for (left) subsystems one and (right) two.

The multidisciplinary optimisation strategy introduces a change in the objectives that are active in each population set. Table 6.57 contains the active objectives present in the system and subsystem pathways and the population sets contained within them, with the system level mirroring the default physical level problem with stiffness kx error and von Mises stress objectives, subsystem one solely focusing on the stiffness kx error objective and finally subsystem two contains the von Mises stress objective and a new objective on minimizing the displacement along the axis of excitation.

The previous multidisciplinary optimisation strategies were applied to design problems which contained a large number of functional evaluations. This allowed the population set sizes for both the system and subsystem pathways to remain the same with little disruption to the overall design process. However for the physical level design problem the numbers of functional evaluations are a tenth of those in the previous design problems, and the transfer of population sets from the system level to subsystem level and vice versa reduce the number of functional evaluations used for search further. Therefore the population set sizes for the subsystem levels have been reduced to allow more search to be undertaken at each subsystem though with an increased selection pressure as a result of a much lower population size. Table 6.58 lists the changes to the population sizes for the system and subsystem levels.

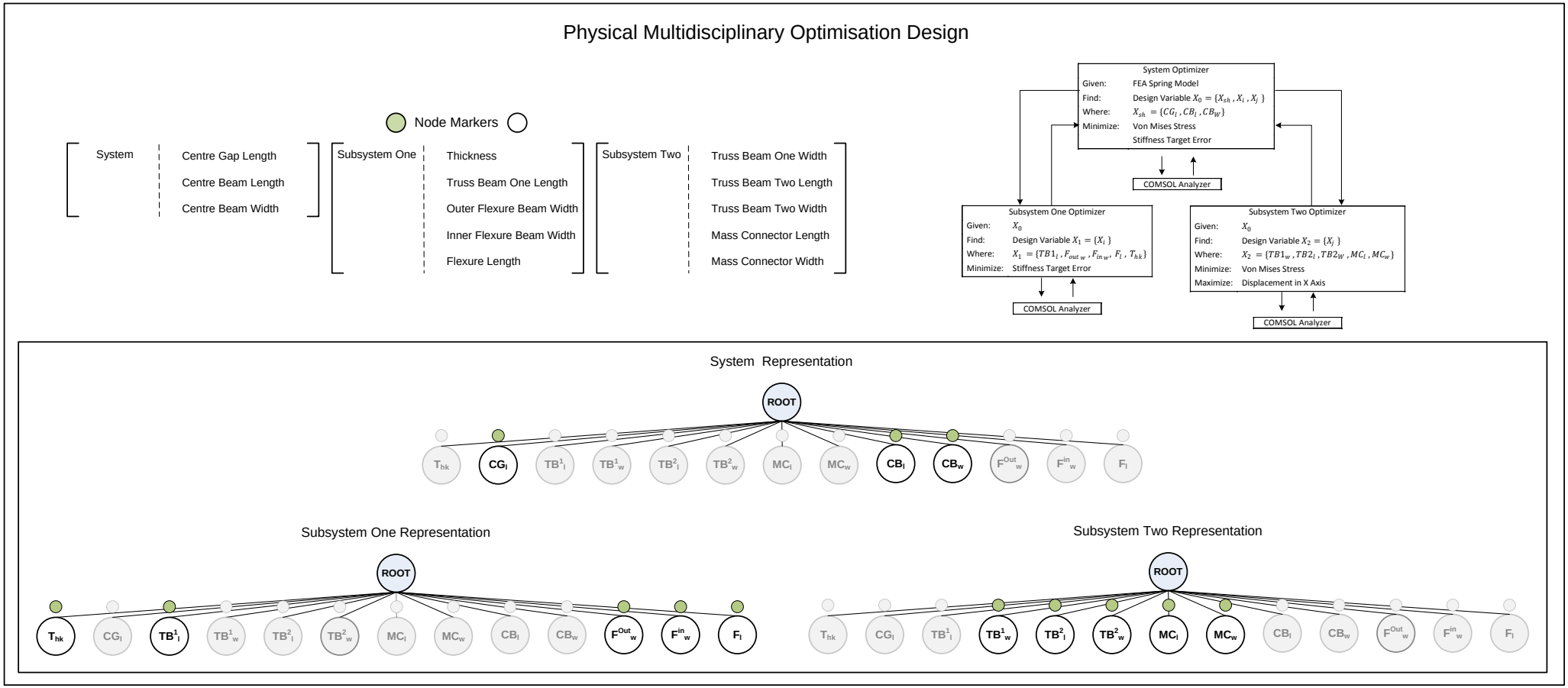


Figure 6.97 Physical Multidisciplinary Optimisation design template, with overview of problem, default representations, and node markers.

Table 6.57 Physical Multidisciplinary Optimisation Objectives

	System Level		Subsystem 1	Subsystem 2	
	Objective 1	Objective 2	Objective 1	Objective 1	Objective 2
Objective Type	Minimize	Minimize	Minimize	Minimize	Minimize
Objective Description	Stiffness Kx Error	Von Mises Stress	Stiffness Kx Error	von Mises Stress	Displacement in X Axis

The multidisciplinary optimisation strategy is outlined in figure 6.97 and consists of a single system and two subsystem pathways. Their associated representations and node markers are included with subsystem one containing decision variables related to the stiffness kx error objective while subsystem two contains variables related to the von Mises stress objective.

Table 6.58 Physical Level Multidisciplinary Algorithmic Parameter Changes

Algorithm Parameter	Default Value
System Population Size	50
System Offspring Size	25
System Selection Size	50
System Replacement Size	75
Grand Pareto Size	150
Subsystem Population Size	25
Subsystem Offspring Size	25
Subsystem Selection Size	25
Subsystem Replacement Size	50
Subsystem Total Size	50

6.5.7 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 6.98 and fully in appendix C.3. A list of the best results ranked by von Mises stress objective values and constrained to having a stiffness error of less than 1% from the target for each run are shown in table 6.59 and their phenotypes shown in figure 6.99 and figure 6.100. Finally the hypervolume values for both algorithms are shown in table 6.60, with the best results shaded.

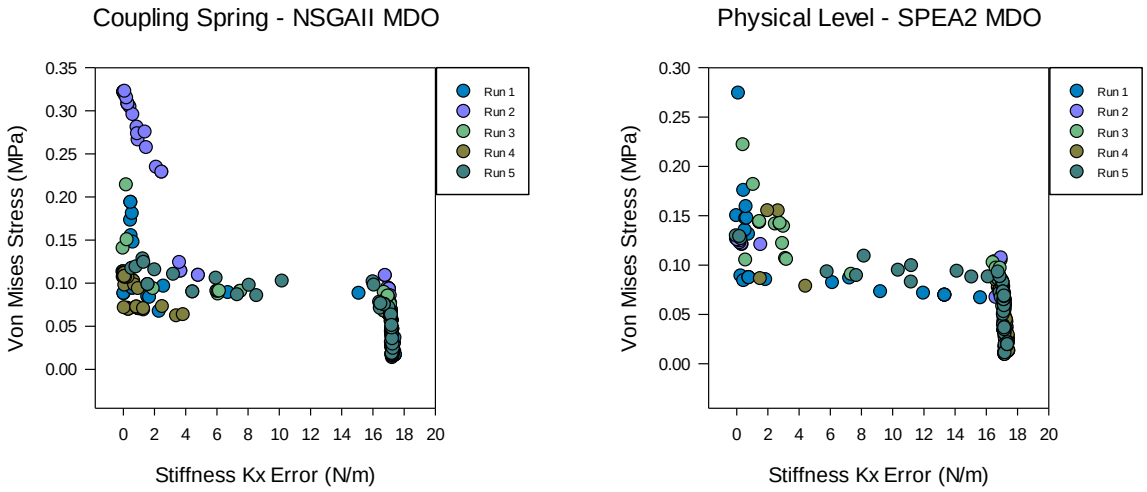


Figure 6.98 Physical level run 1 – 5 final population sets for (left) NSGAII multidisciplinary optimisation and (right) SPEA2 multidisciplinary optimisation

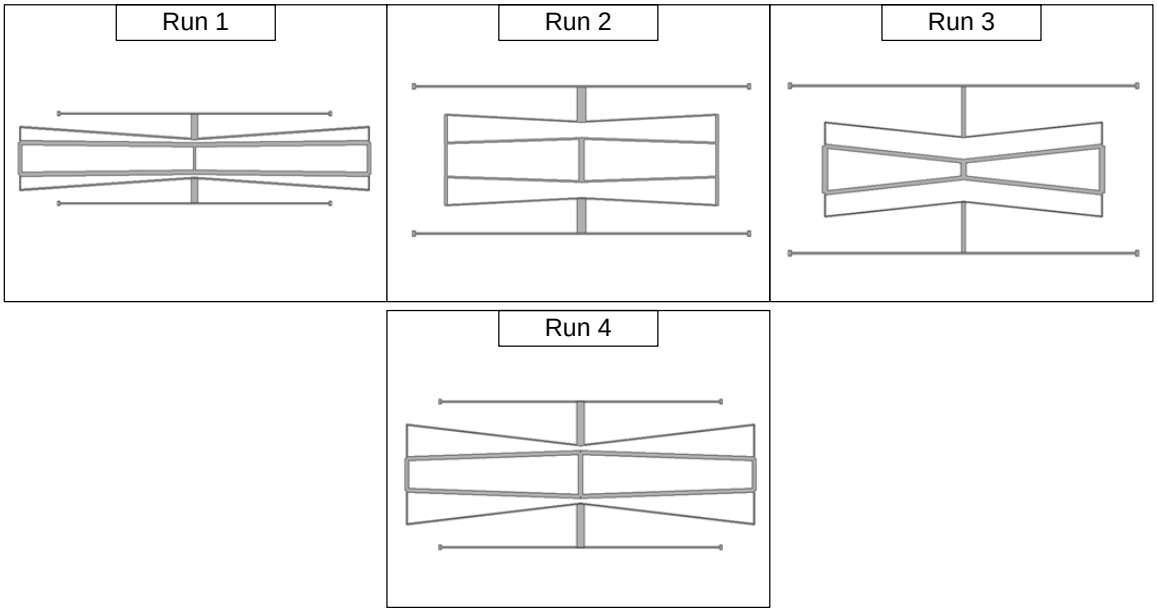


Figure 6.99 Physical level run 1 – 4 coupling spring best result ranked by stiffness kx error for NSGAII multidisciplinary optimization

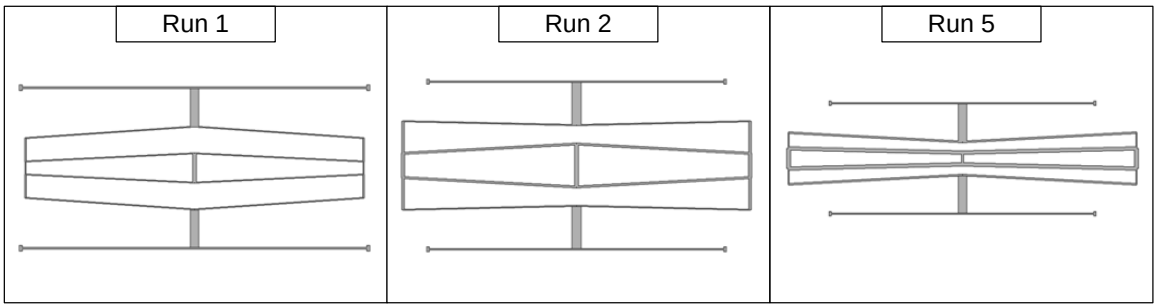


Figure 6.100 Physical level run 1, 2 and 4 coupling spring best result ranked by stiffness kx error for SPEA2 multidisciplinary optimization

Table 6.59 Physical Multidisciplinary Optimisation Best Results Ranked by von Mises Stress for Solutions with ≤ 1% Error

Physical Level NSGAII				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	von Mises Stress (MPa)
1	19	2	0.077681299	0.0872357
2	43	4	0.050454911	0.321148
3	22	3	0.18545279	0.10125073
4	13	2	0.08772584	0.071049777
5	21	0	-	-
Physical Level SPEA2				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Stiffness Kx Error	von Mises Stress (MPa)
1	20	2	0.14346246	0.27380349
2	16	5	0.08707921	0.12513066
3	14	0	-	-
4	14	0	-	-
5	25	1	0.00569731	0.12922860

Table 6.60 Physical Multidisciplinary Optimisation Hypervolume Results for NSGAII and SPEA2

Physical Multidisciplinary Optimisation		
Hypervolume	NSGAII	SPEA2
S^U	18.7993496024206	18.5984053320004
S^M	18.19949331	17.97145709
S^L	17.6388343900102	17.181328919291

* (S^U S^M S^L) [20, 1]

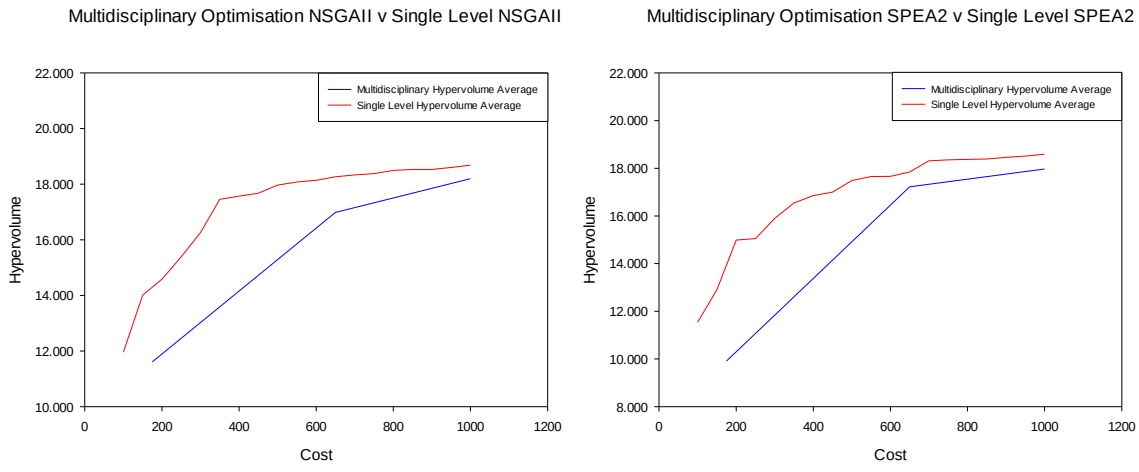


Figure 6.101 Physical level average hypervolume results for the 5 runs of the multidisciplinary optimisation and single Level NSGAII and SPEA2 strategies * (S^U S^M S^L) [20, 1]

Decomposing the coupling spring for the physical level design problem in order to break up the problem into a number of simplified and specific tasks was the aim for applying the multidisciplinary optimisation strategy. Once again both the NSGAII and SPEA2 algorithms were able to produce population sets with similar characteristics to the single level method and also find solutions which contained a synthesis error of less than 1%. The overall hypervolume performance was under par with the single level strategy in both mean and bound values and over the design process as seen in figure 6.101. Here only three population set hypervolume values are taken as the system level is only invoked three times, but the result is clear with more

computational resources required to reach hypervolume values found much early in the single level design process.

Looking closer at each subsystem and their ability to improve coupling spring performance on the stiffness kx error and von Mises stress objectives in figure 6.102 helps to pinpoint one possible reason for the lowered hypervolume performance.

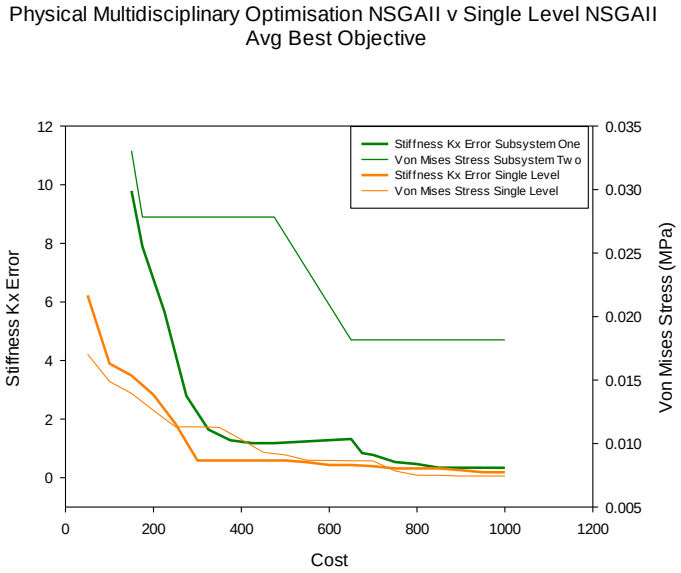


Figure 6.102 Average best objective generational values for multidisciplinary optimisation NSGAII v single level NSGAII for stiffness kx error and von Mises stress objectives

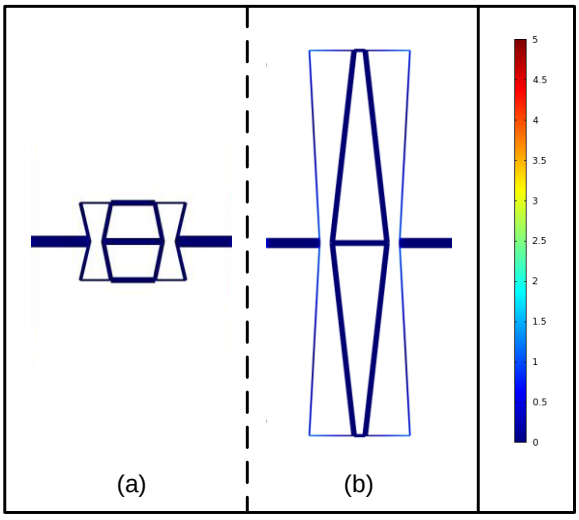


Figure 6.103 Von Mises stress values for displaced coupling springs for subsystem two (a) and subsystem one (b) solutions

Here a comparison is made between the single level design process and the multidisciplinary optimisation strategy, with plots of the average value for the best result found over the design process for each objective in subsystem one, two and the single level populations. The optimisation of the stiffness kx error objective by the multidisciplinary optimisation strategy is similar in performance to that of the single level, with perhaps some lag as a result of the

computational cost spent on other levels early on in the design process. This is in stark contrast with the von Mises objective with subsystem two of the multidisciplinary optimisation strategy failing to match the performance produced by the single level strategy and overall producing a best solution on average with around three times the stress levels found by the single level strategy. The reduced ability within subsystem two to evolve solutions with stress levels that match the single level performance is no doubt a contributing factor to the lower hypervolume performance.

The behaviour of subsystem two, and the overall design process was greatly affected by the phenotypes evolved and retained throughout, in particular those within subsystem one. Figure 6.103 shows two polar examples in both the stiffness kx error and von Mises stress objectives. As discussed in section 6.5.1 the shape and sizing of the coupling spring effect their behaviour, with small compact structures having a reduced von Mises stress, though at a cost of a larger stiffness error, and large flexure springs the opposite of this, normally evolved to match the stiffness target. Phenotypes which contain larger flexure beams, often with small outer widths, have concentrations along there outer flexures which contribute to the overall von Mises stress values the most, while the inner core contributes a much smaller share. This is due to the deformation and torsion of the outer flexures, a direct result of the applied force and the sizing values taken, where as in the case of the more compact rigid structure any force is simply transferred with little to no deformation.

The frequency of solutions within both subsystem populations which contain large flexure beams such as those shown in figure 6.103b is much larger as a result of one particular factor, subsystem one containing only a single objective, that for minimizing the stiffness kx error. This results in the culling of solutions with a large error very rapidly over the design process and also retaining solutions which provide a lower error, in this instance long flexure beams. Subsystem one controls the flexure components through the multidisciplinary decomposition strategy and therefore any new offspring will have these long flexures passed to subsystem two through the geneswapper module. This makes evolving compact structures more difficult as the main driver for evolving the flexures is subsystem one which has no selection pressure towards evolving these types of rigid design. The one benefit of such an approach is that it forces subsystem two to evolve these longer flexure structures to have lower stress levels, a desirable feature which matches the design problem objectives.

One of the characteristics of the multidisciplinary optimisation strategy is the ability to focus populations within different subsystems to evolve different components of the MEMS devices towards different and more specified objectives to that of the whole design problem. The combination of the various subsystem genotypes may provide solutions which as a whole benefit in regards to the system level design problem. Figure 6.104 gives a simple example from run 4 of the NSGAII multidisciplinary optimisation strategy on how such a combination of components within the coupling spring can benefit the solution, in this instance a precursor to the best solution found as seen in figure 6.97.

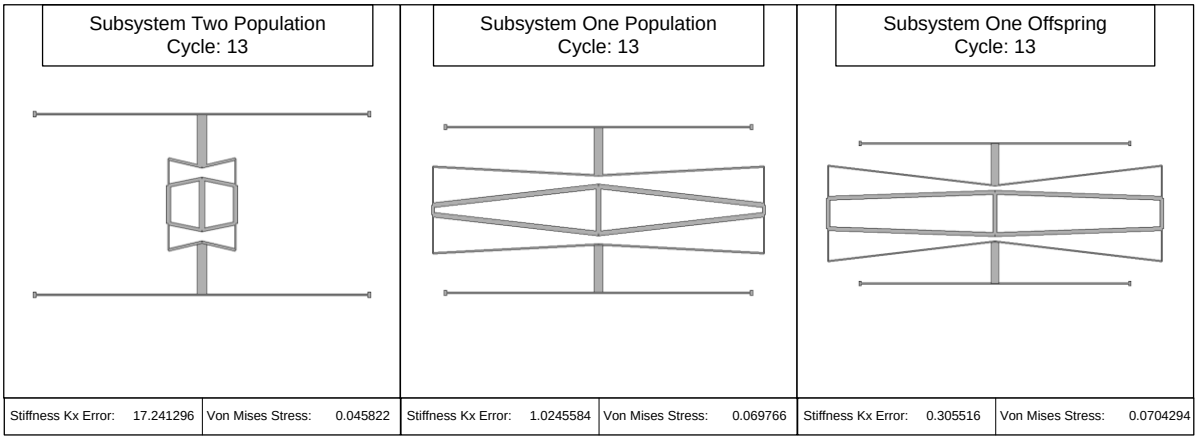


Figure 6.104 Generational phenotypes for multidisciplinary optimisation NSGAII run 4

6.5.8 Physical level comparison and analysis

Solid modelling of MEMS devices and further analysis using computational expensive FEA or fluidic analysis is becoming more important in the final stages of design synthesis and optimisation of MEMS. This is only going to increase as more disciplines are opened up to the application of MEMS and the requirement to undergo single or coupled analysis from multiple disciplines requires the use of physical level modelling and analysis tools.

One of the disadvantages of undertaking automated design optimisation of solid models particularly within the field of engineering is the cost of evaluating any designs that are created. Building solid models, meshing them using finite element method and then performing analysis is expensive in both computational resources such as memory and the cost it takes in wall clock time to perform the analysis.

Table 6.61 Physical Level Hypervolume Results for Single and Multi-level Strategies for Both NSGAII and SPEA2 algorithms.

Physical Level NSGAII				
Hypervolume	Single Level	Multi-Level Evaluation	Multi-Level Parameterization	Multidisciplinary Optimisation
S^U	19.009083302527	18.7052154228448	18.7754454455268	18.7993496024206
S^M	18.6830137	18.2432254	16.53080026	18.19949331
S^L	18.3231287492148	17.3919364330269	8.70650139026446	17.6388343900102
Physical Level SPEA2				
Hypervolume	Single Level	Multi-Level Evaluation	Multi-Level Parameterization	Multidisciplinary Optimisation
S^U	18.780618043083	18.94877986031	19.1241410324346	18.5984053320004
S^M	18.5908500	18.6494235	18.07053509	17.97145709
S^L	18.3778343324101	18.193898408987	16.2150877162596	17.181328919291

* (S^U S^M S^L) [20, 1]

Table 6.60 holds the hypervolume performance for the single and multi-level strategies for both NSGAII and SPEA2 algorithms. The application of the multi-level strategies have overall probably had modest to no impact in terms of improving the hypevolume performance, with the single level NSGAII performing the best and similarly SPEA2 performing better if not equal with the other multi-level strategies applied to solve the physical level design problem. This may be slightly disingenuous due to the low availability of functional evaluations and the disproportionate effect this may have on a series of multi-level strategies which require a

somewhat inefficient structure. The effect of migration and movement of solutions / populations into other levels or subsystems requires re-analysis of prior designs which don't add to any further exploration but still use up much needed functional evaluations.

Also each strategy is designed for a specific task or goal in hand, be it lowering computational cost, simplifying the design problem or re-organising it to benefit the design process as a whole. The effect of each has still been shown in each of the strategies employed.

6.6 Summary

The application of the current state of the art in automated MEMS design synthesis and optimisation along with a new set of multi-level strategies have been successfully applied to a number of uni level design problems spread throughout the design hierarchy of MEMS. The automated design synthesis of a system level MEMS bandpass filter problem was greatly improved through the use of a number of multi-level and multidisciplinary strategies, with an increase in the number of successful bandpass filter designs and improvement to their overall filter transmission shapes when compared with the more current state of the art single level MOEAs.

A device level MEMS folded flexure resonator design problem also highlighted the advantages of applying both multi-level and multidisciplinary optimisation strategies with improved performance over the standard single level MOEAs. The application of a multi-level evaluation strategy was able to significantly increase the number of functional evaluations available to the optimizer and as a result allowed them to produce superior design in terms of cost reduction (total area) over the current state of the art single level MOEAs. The application of a multi-level parameterization strategy both increased the number and variety of designs evolved by the MOEA optimizers and the ability to structure the topology and sizing of the models used allowed for the more rigid and traditional folded flexure design to be gradually improved over the design process. Finally breaking up the modelling and analysis of the folded flexure device into a number of functional units in a MDO strategy improved the design synthesis for the central mass and folded flexure springs of the device and aided the overall design process.

The last uni level design example fell upon the physical level of MEMS modelling and analysis through the use of FEA of a MEMS 'butterfly' coupling spring. This proved to be a more complex problem due in part to the reduction in the available functional evaluations for the optimizers. The single level MOEAS provided some of the more successful designs with the multi-level design strategies at best matching their performance. However each strategy in some way aided the design process, through increased functional evaluations to simplifying the design process through a reduction in the number of design variables open to the optimizer.

The uni level consists of design problems which are restricted to a single level of modelling and analysis within the MEMS design hierarchy. The next chapter looks to expand this to two levels of modelling and analysis in a series of coupled bi-level MEMS design problems.

7

Bi-Level Design Optimisation

This chapter shifts the focus from the design synthesis and optimisation of MEMS at a single level of modelling and analysis to a series of design problems which couple two levels together. The first of these design problems consists of a system and device level problem for the design optimisation of a bandpass filter, while the second focuses upon the individual bandpass components in a physical and device level design problem. Once again both the current state of the art algorithms for automated design of MEMS, single level MOEAs, and the multi-level and multidisciplinary optimisation algorithms are applied and compared.

The hierarchical nature of MEMS modelling, analysis and design often leads to the interaction of multiple models from a number of levels during the design synthesis of the MEMS device or component. This can naturally stem from the fact that more and more disciplines are being added to the field of MEMS and different modelling and analysis tool are required to undertake the design optimisation of such coupled design problems.

The bi level design optimisation chapter focuses upon such coupled design problems within MEMS synthesis, where more than one modelling and analysis level is used for the design of a MEMS device and in certain circumstances required to do so. Both in chapter 4 and within the previous uni level design examples there exist the circuit level electrical discipline derived from the electrical equivalent model of the bandpass filter along with the mechanical and structural disciplines invoked from the device level layout of the folded flexure resonator components that make up the filter device. These two levels are components are combined in what is a coupled system – device level MEMS design problem, and form a more complex example of bandpass filter design.

The final bi level MEMS design problem looks to couple together the physical models used for design optimisation, in particular the 2D layout and 3D solid modelling found in the device and physical hierarchical levels respectively. Both a folded flexure resonator and butterfly coupling spring have been explored individually as separate MEMS design optimisation problems. However in the total MEMS bandpass filter system both the folded flexure resonators and coupling springs are coupled together physically and their function is heavily influenced by either component. Therefore as shown in figure 7.1 two coupled MEMS design problems, a system – device and a device – physical problem are explored with both the standard single level MOEA

strategy and the new multi-level and multidisciplinary optimisation strategies applied to solve them.

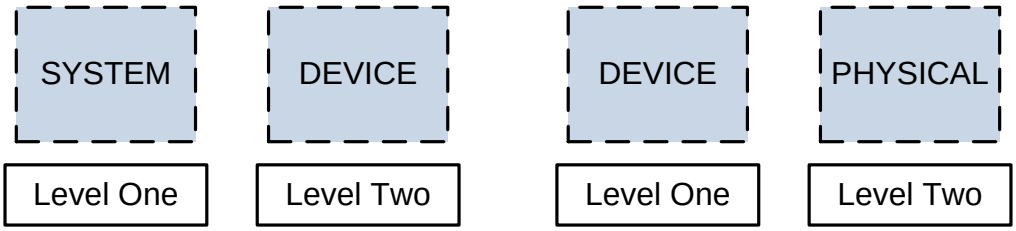


Figure 7.1 Bi level design optimisation of MEMS using multi-level design strategies

7.1 System – Device level design optimisation

The main mechanical components of the MEMS bandpass filter outlined in this study are the resonators that make up the individual frequency responses with the frequency transmission shape and the soft coupling springs that couple them together.

Both chapters 5 and 6 explored how each of the components can be optimised separately so as to allow MEMS bandpass filters to be synthesised automatically at the system and device levels. The approach used the characteristic ability to model the bandpass filter in both the electrical and mechanical domain so its function could be described and optimised as an electrical equivalent model and then later fabricated as a 2D layout using a mechanical model representation as target design goals. This particular methodology for design optimisation was classed as an 'de-coupled' approach, with their being a loose link between both the system and device levels through the electrical-mechanical conversion, but with both levels being optimised in separate 'events'. However it is possible to undertake a 'coupled' design optimisation of the bandpass filter through a role reversal with the synthesis and optimisation of the bandpass filter at the device level and at the same time converting the structure into an electrical equivalent model for analysis at the system level as shown in figure 7.2.

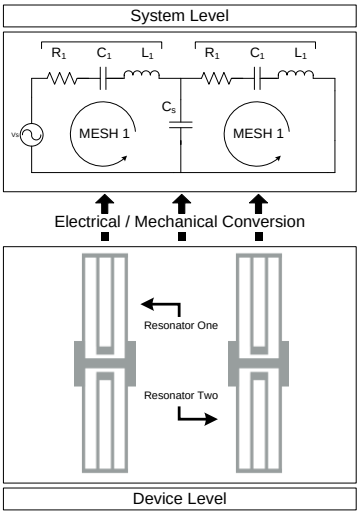


Figure 7.2 Mechanical to electrical conversion for MEMS bandpass filter system – device level problem

Some of the advantages of a 'coupled' methodology include a finalised and feasible 2D layout designs for the MEMS bandpass filter. In the previous 'de-coupled' approach after system level optimisation was completed only design target values are extracted from the electrical equivalent model, and it is a possibility that such values cannot be successfully realized into actual 2D layout designs because their values are simply unattainable. This is not the case with the 'coupled' methodology as the electrical equivalent values are taken from fully realized 2D layout designs already evolved. The evolution of parameterized 2D layout models also removes one of the difficulties associated with the system level electrical model, that being the starting parameter bounds for the decision variables and choosing correct values so the initialisation population lies within the frequency range of interest. The more open 2D layout decision variables allow for a much larger range of mass and stiffness and therefore inductance and capacitance values for each 'RCL' tank.

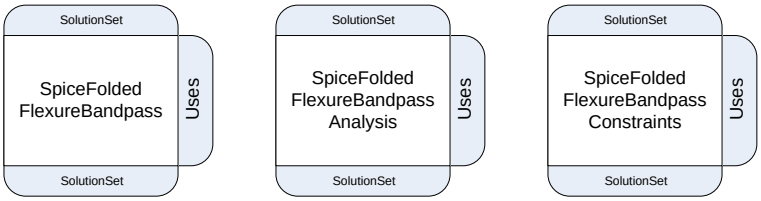


Figure 7.3 System - Device level modules for (a) spice folded flexure bandpass problem information, (b) spice folded flexure bandpass analysis and (c) spice folded flexure bandpass constraints.

Table 7.1 System - Device Level Filter Problem Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Voltage	Real Valued	1	200
Tank Number	Integer	1	9
Finger Number	Integer	1	200
Comb Thickness (μm)	Real Valued	2	30
Coupling Spring Capacitance (F)	Real Valued	3E-15	8E-15
Central Mass Width (μm)	Real Valued	10	400
Shuttle Yolk Length (μm)	Real Valued	2	400
Shuttle Yolk Width (μm)	Real Valued	10	400
Beam Length (μm)	Real Valued	5	100
Beam Width (μm)	Real Valued	2	20
Truss Length (μm)	Real Valued	0	400
Truss Width (μm)	Real Valued	2	20
Thickness (μm)	Real Valued	2	200
Objectives		Constraints	
Bandpass Filter Response Error	Minimize	Tank Constraint 1	
Bandpass Central Frequency Error	Minimize	[Shuttle Yolk Length - Central Mass Width]	Inequality > 0
		Tank Constraint 2	
		[(([Beam Width] * 4) - ([Truss Length] * 3)) + 9e-6]	Inequality < 0

The coupled system – device level design problem requires the construction of new problem specific modules which need to be incorporated into the design framework. The new modules

are shown in figure 7.3 and include the spice folded flexure bandpass problem information, analysis and constraint functions required to undertake the design optimisation at the system – device level.

The overall system - device level design problem is shown in figure 7.4 along with the structure and node markers associated with the coupled representation. The representation variable bounds and design problem objectives and constraints are shown in table 7.1. The design problem uses the analytical folded flexure model and therefore does not require the use of the structural crossover module found in the device level problem pathway example. The constraints reflect the analytical model and are applied to each 'tank' component within the representation and a failure in any leads to a constrained design. Algorithmic parameters for both NSGAII and SPEA2 remain unchanged from those used in the previous default uni level design problems, and the total functional evaluations remains 10,000.

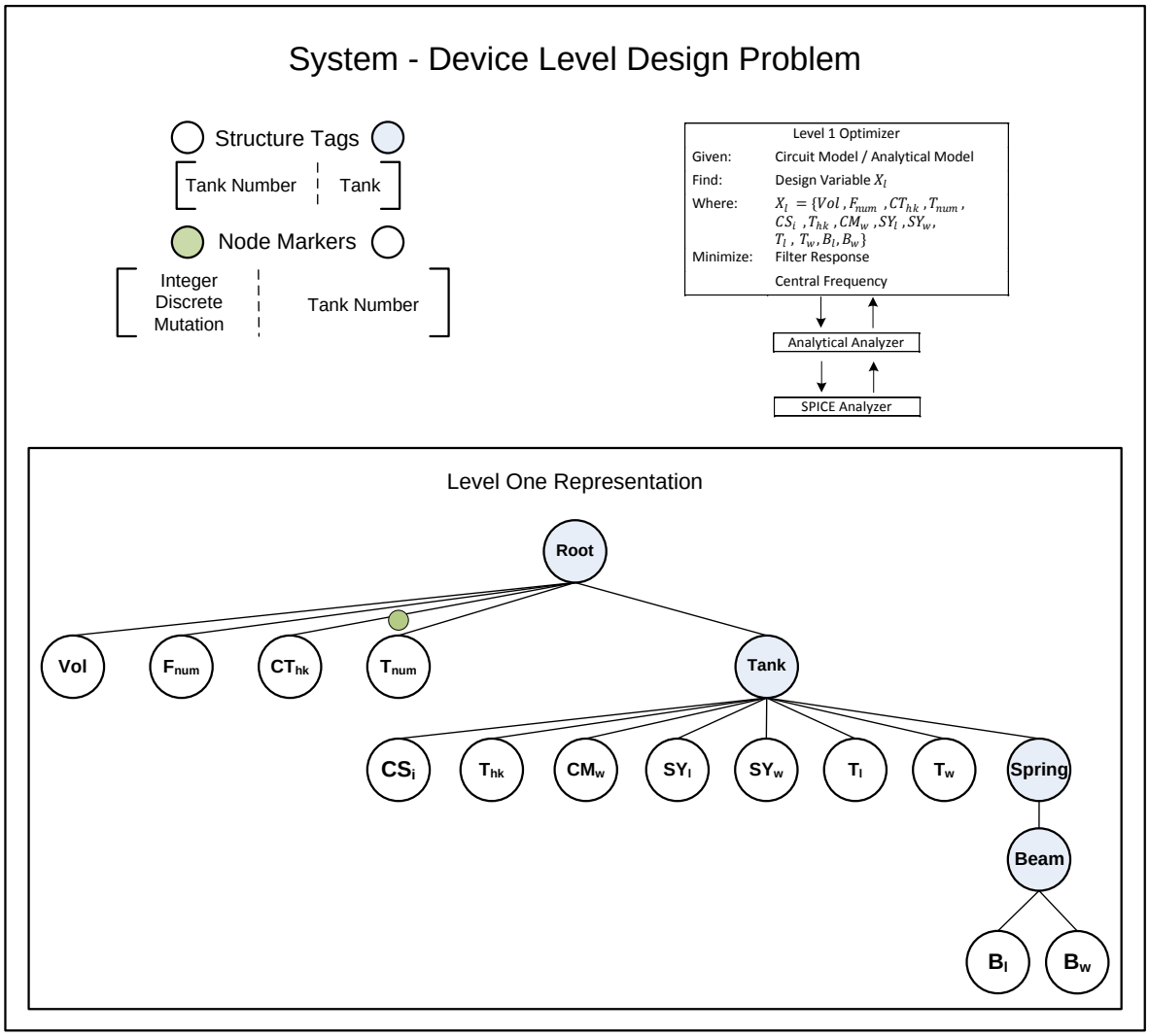


Figure 7.4 System - Device level design template, with overview of problem, default representation, associated structure tags and node markers.

7.1.1 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 7.5 and fully in appendix D.1. Also shown are the best frequency transmissions for each algorithm, ranked by the filter frequency objective in figure 7.6 and fully in appendix D.1 along with their objective values in table 7.2. The mass and stiffness values for these solutions are shown in table 7.3 along with the phenotypes for the single level NSGAII strategy in appendix D.1. Finally the hypervolume values for both algorithms are shown in table 7.4, with the best results shaded.

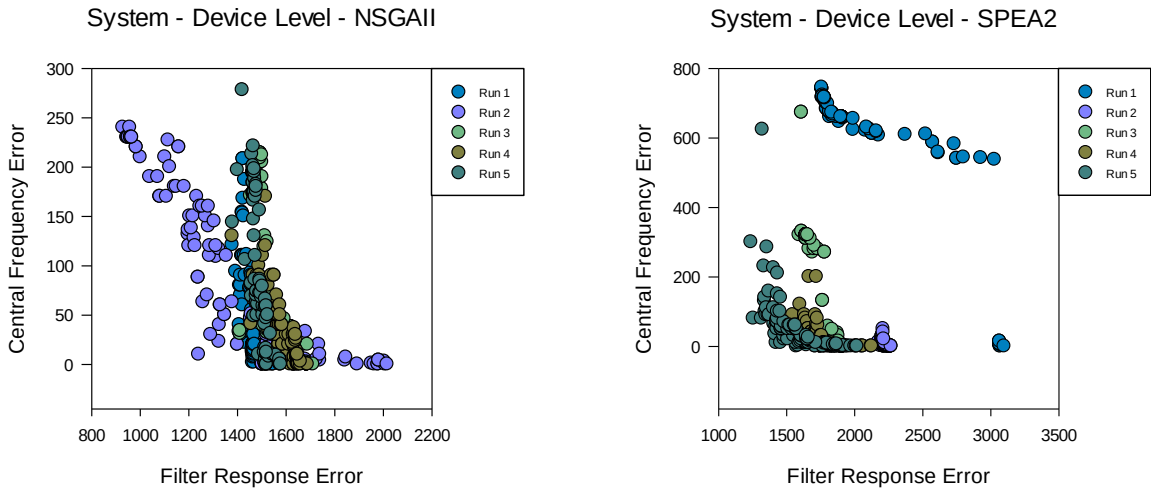


Figure 7.5 System – Device level run 1 – 5 final population sets for (left) NSGAII and (right) SPEA2

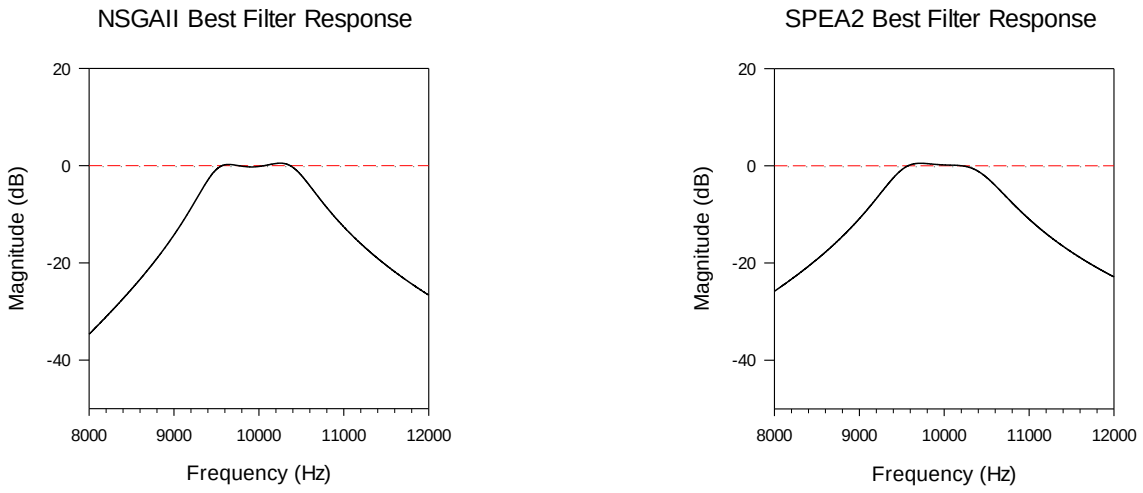


Figure 7.6 System – Device level best filter response ranked by filter frequency objective for (left) NSGAII and (right) SPEA2

The synthesis and optimisation of a MEMS bandpass filter can be achieved through the use of system and device level modelling and analysis tools [55][59][66], often undertaken separately as seen in the previous chapter. However rather than using mechanical values extracted from an evolved electrical equivalent bandpass model to produce a 2D layout design for the bandpass filter an approach which coupled both system and device level modelling and analysis in a single design process was constructed. This particular schema evolved the 2D layout design for the

Table 7.2 System – Device Level Bandpass Filter Results

System – Device Level NSGAII					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	2	1378.996	121	164.919	5
2	9	929.301	240	174.690	9
3	4	1411.410	34	147.375	5
4	3	1379.048	130	175.287	5
5	7	1380.461	144	115.230	3
System – Device Level SPEA2					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	4	1759.440	741	57.352	2
2	1	2176.510	12	177.680	5
3	5	1593.128	320	75.082	3
4	6	1546.416	90	136.135	5
5	2	1239.262	300	114.668	9

Table 7.3 System – Device Level Mechanical Values for Mass and Stiffness for the Best Solutions Ranked by the Filter Error Objective for Both NSGAII and SPEA2 Algorithms.

System – Device Level NSGAII						
Individual Folded Flexure Resonator Values		Experimental Run				
		1	2	3	4	5
Tank 1	Mass (kg)	2.423E-08	1.874E-08	1.771E-08	1.080E-08	5.034E-08
	Stiffness (N/m)	120.627	68.406	55.925	38.183	221.020
Tank 2	Mass (kg)	2.132E-08	1.931E-08	1.337E-08	1.526E-08	4.068E-08
	Stiffness (N/m)	81.724	72.233	63.984	75.400	49.269
Tank 3	Mass (kg)	1.916E-08	2.284E-08	2.294E-08	1.598E-08	5.698E-08
	Stiffness (N/m)	74.828	66.126	69.502	77.264	213.619
Tank 4	Mass (kg)	2.127E-08	2.143E-08	1.170E-08	1.460E-08	-
	Stiffness (N/m)	80.473	77.342	81.829	38.154	-
Tank 5	Mass (kg)	1.668E-08	1.393E-08	2.370E-08	1.689E-08	-
	Stiffness (N/m)	49.029	69.041	98.148	63.289	-
Tank 6	Mass (kg)	-	1.265E-08	-	-	-
	Stiffness (N/m)	-	49.524	-	-	-
Tank 7	Mass (kg)	-	1.692E-08	-	-	-
	Stiffness (N/m)	-	62.917	-	-	-
Tank 8	Mass (kg)	-	1.552E-08	-	-	-
	Stiffness (N/m)	-	86.615	-	-	-
Tank 9	Mass (kg)	-	2.168E-08	-	-	-
	Stiffness (N/m)	-	82.824	-	-	-
System – Device Level SPEA2						
Individual Folded Flexure Resonator Values		Experimental Run				
		1	2	3	4	5
Tank 1	Mass (kg)	9.834E-08	7.614E-09	1.073E-07	2.223E-08	1.201E-09
	Stiffness (N/m)	331.164	31.805	460.932	106.320	3.181
Tank 2	Mass (kg)	2.000E-08	1.784E-09	6.829E-08	2.688E-08	3.765E-09
	Stiffness (N/m)	84.341	7.890	199.925	99.675	5.125
Tank 3	Mass (kg)	-	6.534E-09	5.283E-08	2.801E-08	2.552E-08
	Stiffness (N/m)	-	25.414	189.470	107.984	99.545
Tank 4	Mass (kg)	-	2.523E-08	-	3.281E-08	3.774E-09
	Stiffness (N/m)	-	88.710	-	75.870	15.887
Tank 5	Mass (kg)	-	5.410E-08	-	2.664E-08	3.599E-09
	Stiffness (N/m)	-	76.542	-	108.055	9.785
Tank 6	Mass (kg)	-	-	-	-	6.724E-09
	Stiffness (N/m)	-	-	-	-	30.521
Tank 7	Mass (kg)	-	-	-	-	1.567E-09
	Stiffness (N/m)	-	-	-	-	9.607
Tank 8	Mass (kg)	-	-	-	-	2.331E-08
	Stiffness (N/m)	-	-	-	-	4.695
Tank 9	Mass (kg)	-	-	-	-	8.529E-08
	Stiffness (N/m)	-	-	-	-	370.298

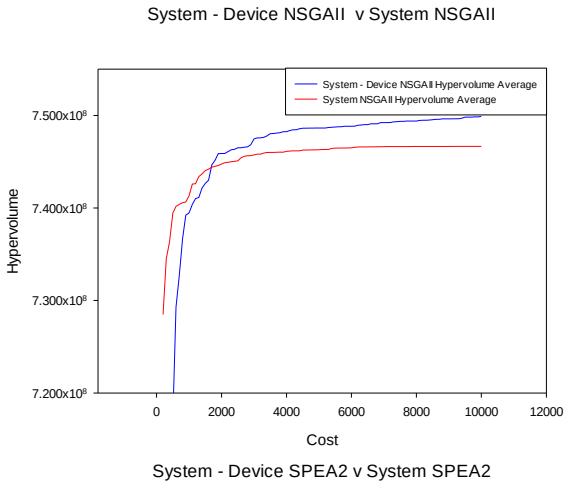
Table 7.4 System - Device Level Hypervolume Results for NSGAII and SPEA2

System – Device Level		
Hypervolume	NSGAII	SPEA2
S^U	45293277.4844393	43783808.3056236
S^M	43500546.71823	41489209.49112
S^L	42936836.5109377	38997795.1789861

* (S^U S^M S^L) [10000, 5000]

bandpass filter first before converting values into an equivalent electrical model which could then be analysed using a circuit analysis tool Spice [55]. The performance for both the NSGAII and SPEA2 algorithm in evolving filter transmission shapes which match the target goals is mixed with an interesting switch in algorithmic performance with NSGAII now performing better than SPEA2 compared with the system level design problem.

The generational hypervolume performance of both the system – device level and system level methodologies for evolving bandpass filters are shown in figure 7.7. The performance for NSGAII is a considerable improvement over the system level approach while there is a slight decrease in performance for SPEA2. This is unsurprising as the system level results for SPEA2 were the best set of the two algorithms while the system – device level SPEA2 set was the worst performing of the two. Another interesting characteristic of the system – device level performance is the trajectory for improvement in hypervolume values show a positive linear trend while the system level results show convergence at the end of the design process.



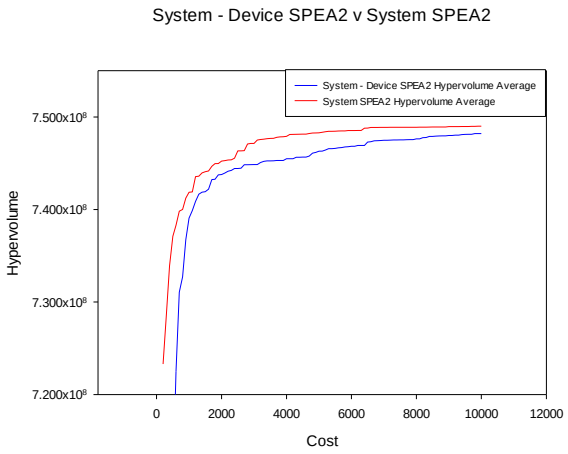


Figure 7.7 System - Device Level and system level average hypervolume results for the 5 runs of the single level NSGAII and SPEA2 strategies * ($S^U S^M S^L$) [182000, 4150]

The lack of convergence of the hypervolume performance is mirrored with the lack of convergence of the population sets evolved by both algorithms over the ten experimental runs as seen in figures 7.5 and appendix figures D.1 and D.2. With each mechanical resonator tank having four times as many design variables as the electrical equivalent 'RCL' model there is a much higher chance of variation in the phenotype of each solution. The variation in phenotype can also be effected by how much a small change has on the structure of each individual resonator in comparison with a simple variation of a component of the 'RCL' tank in the electrical model. The range of variation in mass and stiffness and equivalent inductance and capacitance values of each tank is more likely to be larger for the mechanical resonator model than the through direct variation of the 'RCL' tank values. Figure 7.8 highlights the effect of such variation by taking the best solution found in run 1 for NSGAII and then applying polynomial mutation with a 5% invocation over a thousand separate runs and plotting the effect on filter frequency phenotype. The mutation is applied to a mechanical model and specifically to the variables associated with each mechanical resonator tank, or an electrical equivalent model and the 'inductance', 'capacitance' and 'coupling spring capacitance' variables for each tank. Both models show a large affinity of solutions which remain unchanged or have little change to the filter frequency error objective, however where the electrical model has over 600 out of 1000 variations within this band, the mechanical model has a much smaller proportion. This is in addition to what is a clear gradient of descent of varied solutions from this peak for the mechanical model highlighting how much more polynomial mutation can affect the frequency error phenotype for the mechanical model than the electrical one.

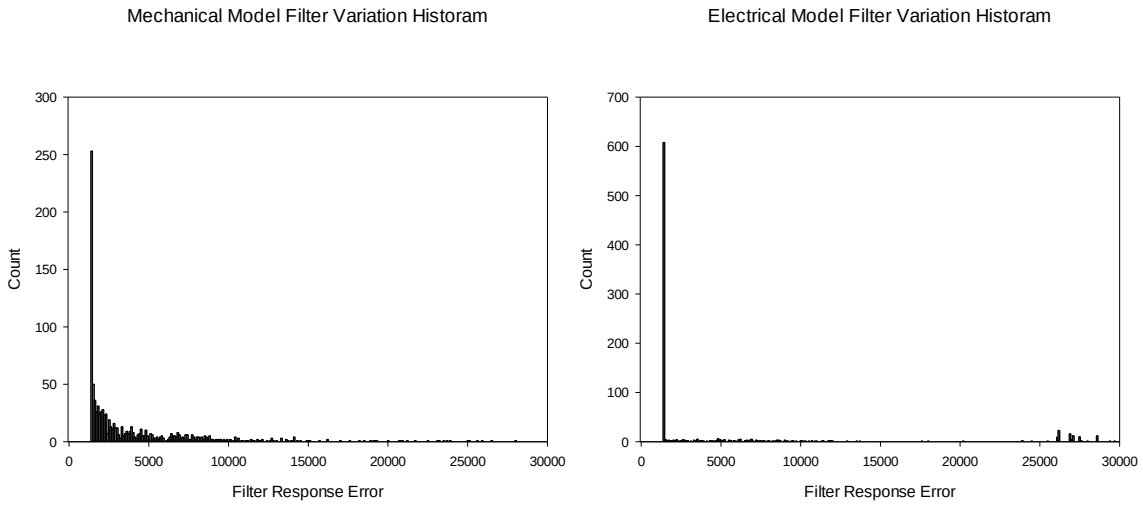


Figure 7.8 Histogram for mechanical and electrical filter model variation for best solution NSGAII run 1

Table 7.5 System Level Mechanical Values for Mass and Stiffness for the Best Solution Ranked by the Filter Error Objective SPEA2 run 4.

System Level SPEA2 Run 4	
Individual Folded Flexure Resonator Values	Experimental Run 4

Tank 1	Mass (kg)	1.865E-09
	Stiffness (N/m)	8.2999
Tank 2	Mass (kg)	2.739E-09
	Stiffness (N/m)	10.884
Tank 3	Mass (kg)	2.205E-09
	Stiffness (N/m)	7.8543

A number of other characteristic changes to the solutions found at the outcome of the system – device level design process have also occurred. Firstly the change in filter transmission shape for the best solutions found as seen in figures 7.6 and appendix figures D.3 and D.4 in comparison with the system level design process. Here the filter transmission shapes contain a much flatter pass band with little to no error but with a trade off of a shallower drop off and as a result higher stop band error. The change in the transmission shape, in particular in relation to the drop off is interesting considering another characteristic shift from the system level design problem in finding solutions which contained more than three resonator tanks within the bandpass filter. Table 7.3 holds the mechanical values for each tank of the best bandpass filter designs found for both NSGAII and SPEA2 and in both examples the increase in tank number is apparent, however interestingly an increase in drop off one might expect through additional tanks is not. The effect of a structural representation and the ability to add or remove tanks during the variation operators is also apparent, with a number of filter topologies contain resonator designs with similar shape, sizing and mechanical values.

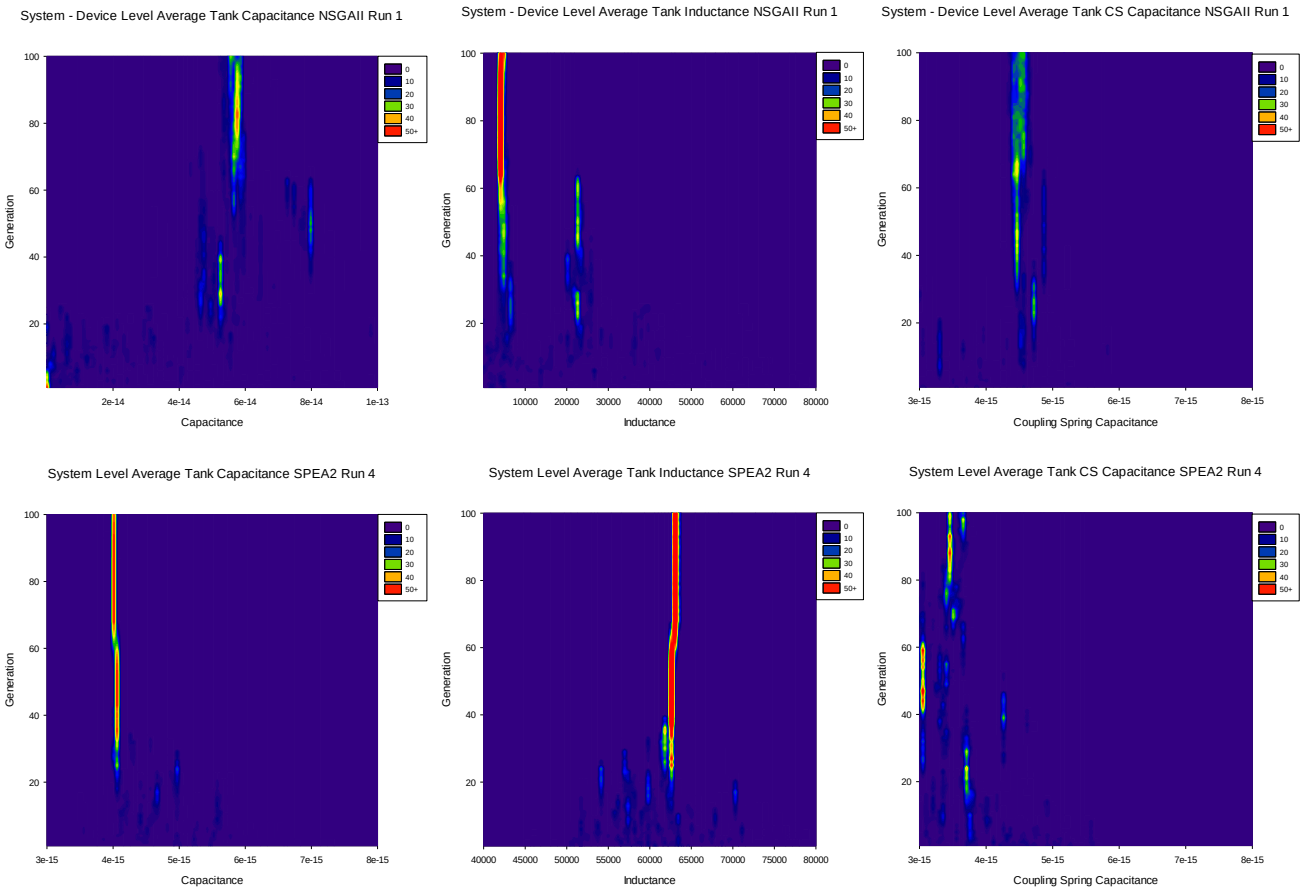


Figure 7.9 Three generational histogram plots for system – device Level NSGAII run 1 and system level SPEA2 run 4 plotting average tank values for capacitance (left), inductance (middle) and coupling spring capacitance (right) against population density.

It is hard to ascribe any particular factor which may affect transmission shape in the way the system – device level methodology has, however one explanation may be where the system – device or system level phenotypes lie within the design search space. Table 7.5 holds the mechanical values for the best solution found by system level SPEA2 run 4, with resonators containing mainly soft stiffness values and smaller mass values in comparison to those found by the system – device level methodology. The ability for the mechanical representation used within the system – device level problem to provide more variation in the 'RCL' electrical equivalent model values can result in more alternative phenotypes like the best solutions evolved at the end of each run shown in figure 7.6 and appendix figures D.3 and D.4. Figure 7.9 highlights the difference in the phenotype space each methodology searches when evolving solutions that give the required transmission frequency response targeted by the designer. Here run 1 of the system – device level NSGAII approach and run 4 of the system level SPEA2 approach are shown with their generational histogram density plots for the average values of each resonator tank for capacitance, inductance and coupling spring capacitance over the entire population. The system – device level methodology with its focus on evolving the 2D layout first concentrates the capacitance and inductance values in different regions of the search space, in particular with inductance being values a tenth of those evolved by the system level method.

7.1.2 Multi-level evaluation

The application of multiple levels of evaluation continues with the system – device level design problem in much the same fashion as the previous system level example. The same three levels of evaluation are used with the first two only sampling a third of the data points and the final level providing the default level of analysis. Figure 7.2.1 outlines the system – device multi-level evaluation design problem with both the representation and node markers used through all three levels.

Four migrator modules are utilized in this particular design strategy to allow individuals to move between neighbouring levels. The values for migration percentage along with the cycle count when migration is invoked are shown in table 7.2.1.

Table 7.6 Migrator Module Parameters for Multi-Level Evaluation

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	20	4
Level 2	Level 1	20	4
Level 2	Level 3	20	4
Level 4	Level 3	20	4

The objectives for design synthesis and optimisation remain the same as in the previous system – device level problem, and the multi-objective evolutionary algorithms used for this design problem are NSGAII and SPEA2. The number of functional evaluations remains at 10,000, however the cost of each evaluation is different depending on the level it is undertaken. As in

the system multi-level evaluation design problem the cost ratios for analysis remain the same as the additional cost of converting the analytical folded flexure model into the electrical equivalent model is negligible.

Table 7.7 Evaluation Cost for SPICE Electrical Equivalent Model

Full SPICE Analysis		Reduced SPICE Analysis	
Frequency Range	Sampling Size	Frequency Range	Sampling Size
1Hz – 15kHz	15,000	1Hz – 15kHz	5,000
Mean Analysis Time (20 Calls)		Mean Analysis Time (20 Calls)	
0.769344005		0.41932398	
Ratio			
1 : 1.834724561			

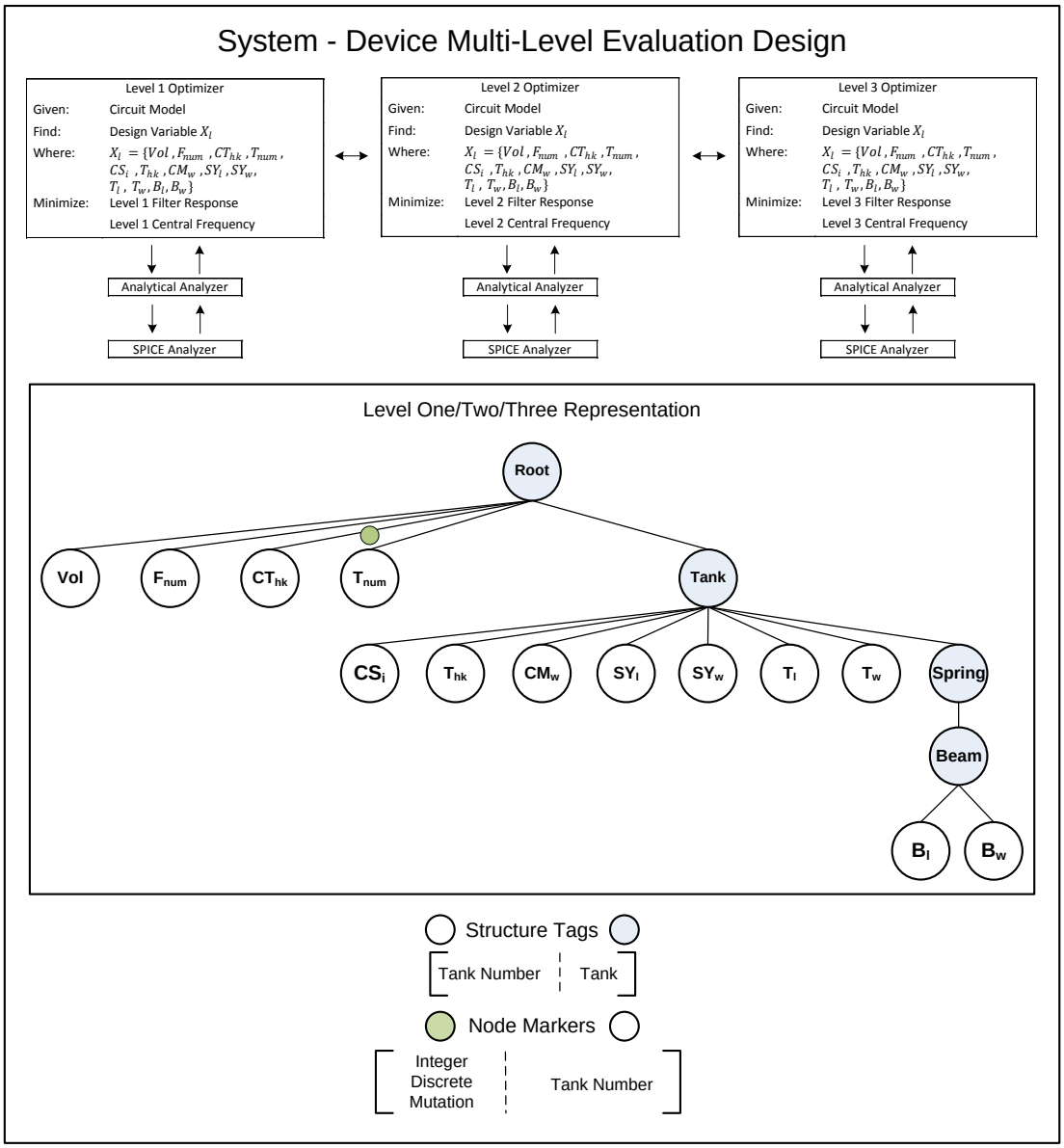


Figure 7.10 System - Device multi-level evaluation design template, with overview of problem, default representation, associated structure tags and node markers.

7.1.3 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 7.11 and fully in appendix D.1. Also shown are the best frequency transmissions for each algorithm, ranked by the filter frequency objective in figure 7.11 and fully in appendix D.1 along with their objective values in table 7.9. The mass and stiffness values for these solutions are shown in table 7.10 along with the phenotypes for the NSGAII strategy in appendix D.1. Finally the hypervolume values for both algorithms are shown in table 7.8, with the best results shaded.

Table 7.8 System - Device Level Hypervolume Results for NSGAII and SPEA2 Multi-Level Evaluation

System – Device Multi-Level Evaluation		
Hypervolume	NSGAII	SPEA2
S^U	45231930.4653885	43629486.6022534
S^M	44060613.34740	41920157.36378
S^L	40901705.6449109	40124595.2576051

* (S^U S^M S^L) [10000, 5000]

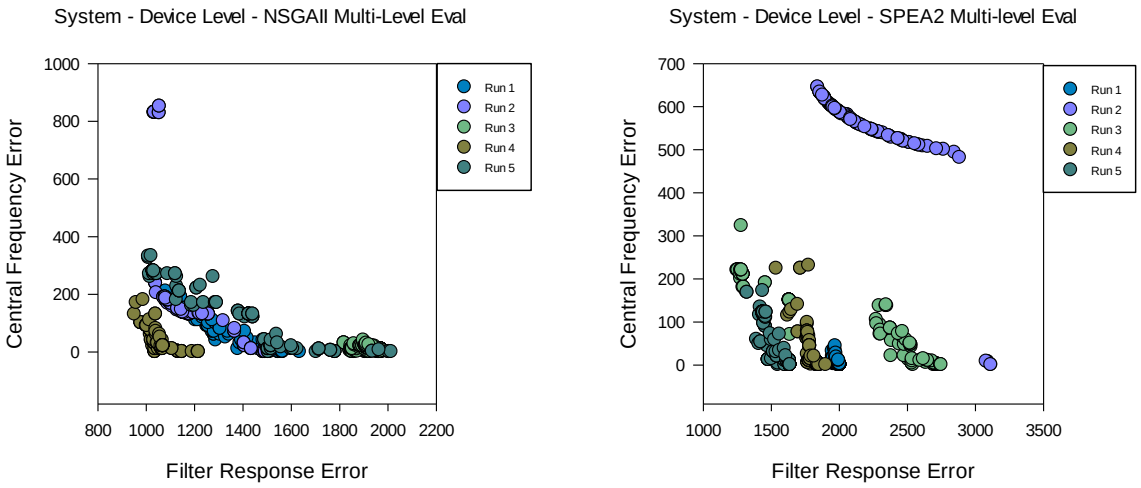


Figure 7.11 System – Device level run 1 – 5 final population sets for (left) NSGAII multi-level evaluation and (right) SPEA2 multi-level evaluation

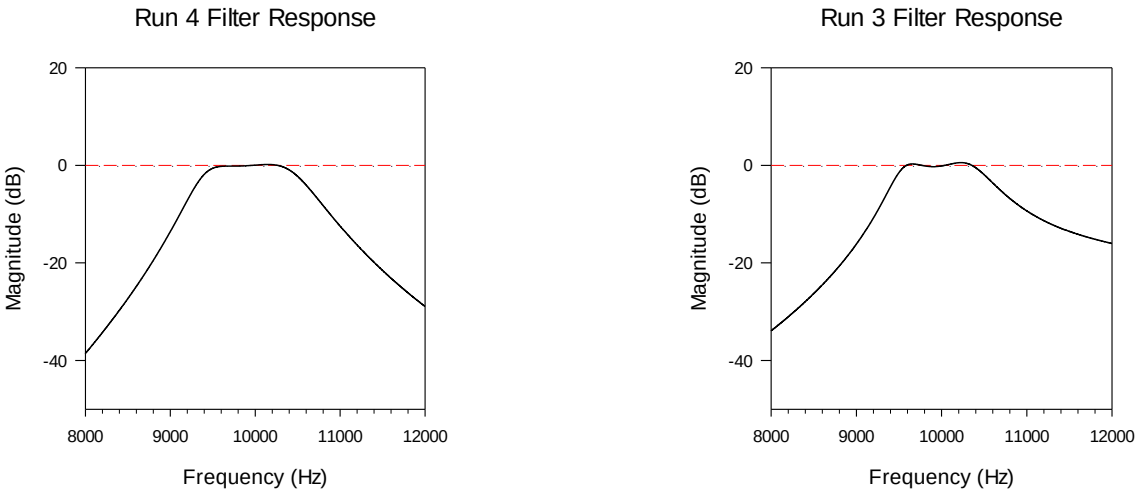


Figure 7.12 System – Device level best filter response ranked by filter frequency objective for (left) NSGAII multi-level evaluation and (right) SPEA2 multi-level evaluation

Table 7.9 System – Device Multi-Level Evaluation Bandpass Filter Results

System – Device Multi-Level Evaluation NSGAII					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	16	1081.396	210	186.55	3
2	40	1032.879	829	183.32	3
3	1	1819.503	30	54.06	3
4	8	952.031	130	152.83	5
5	15	1010.373	331	117.26	4
System – Device Multi-Level Evaluation SPEA2					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	7	1949.871	32	112.55	3
2	75	1842.521	645	154.99	2
3	15	1249.374	220	196.90	3
4	9	1538.098	224	95.01	3
5	4	1324.827	168	44.60	3

Table 7.10 System – Device Multi-Level Evaluation Mechanical Values for Mass and Stiffness for the Best Solutions Ranked by the Filter Error Objective for Both NSGAII and SPEA2 Algorithms.

System – Device Multi-Level Evaluation NSGAII						
Individual Folded Flexure Resonator Values		Experimental Run				
		1	2	3	4	5
Tank 1	Mass (kg)	1.093E-07	9.818E-08	4.313E-08	2.820E-08	2.857E-08
	Stiffness (N/m)	394.611	427.462	182.512	103..425	122.171
Tank 2	Mass (kg)	4.354E-08	1.298E-07	3.430E-08	3.457E-08	9.168E-09
	Stiffness (N/m)	8.531	487.595	26.932	163.100	0.977
Tank 3	Mass (kg)	1.049E-07	1.481E-07	3.858E-08	3.766E-08	2.461E-09
	Stiffness (N/m)	448.998	483.637	145.906	170.139	30.968
Tank 4	Mass (kg)	-	-	-	3.182E-08	4.658E-08
	Stiffness (N/m)	-	-	-	112.699	165.323
Tank 5	Mass (kg)	-	-	-	3.288E-08	-
	Stiffness (N/m)	-	-	-	111.857	-

System – Device Multi-Level Evaluation SPEA2						
Individual Folded Flexure Resonator Values		Experimental Run				
		1	2	3	4	5
Tank 1	Mass (kg)	8.914E-08	1.022E-07	1.548E-07	2.201E-08	1.302E-08
	Stiffness (N/m)	343.726	352.051	553.600	85.599	57.746
Tank 2	Mass (kg)	9.170E-08	1.939E-08	5.289E-08	3.976E-09	9.861E-09
	Stiffness (N/m)	43.691	81.327	346.277	72.394	14.911
Tank 3	Mass (kg)	9.063E-08	-	1.017E-07	2.922E-08	1.305E-08
	Stiffness (N/m)	375.482	-	420.243	102.402	48.431

The application of the multi-level evaluation strategy to the system – device level design problem shows in both table 7.9 and figure 7.13 an improvement in performance over the single level method however in a more subdued effectiveness, especially compared with the performance boost the same strategy had upon the system level design problem. This is perhaps unsurprising given the higher performance at the single level for the system – device level design problem compared with the system level design problem. The characteristics of both the final population sets in figures 7.11 and the associated best filter frequency transmissions in figure 7.12 are similar to the single level method with the exception of the convergence of SPEA2 run 2 population set and an overall lower number of tanks within the final solutions.

Once again the hypervolume performance over the entire design process shows continued improvement even at the end with little sign of convergence as seen in figure 7.13. It is unclear whether continued optimisation would provide better filter frequency transmission solutions or simply allow the optimizer to fill out the Pareto front already attained, however compared with the system level method and its general affliction towards convergence, the system – device multi-level evaluation strategy provides more scope for improvement given additional functional evaluations.

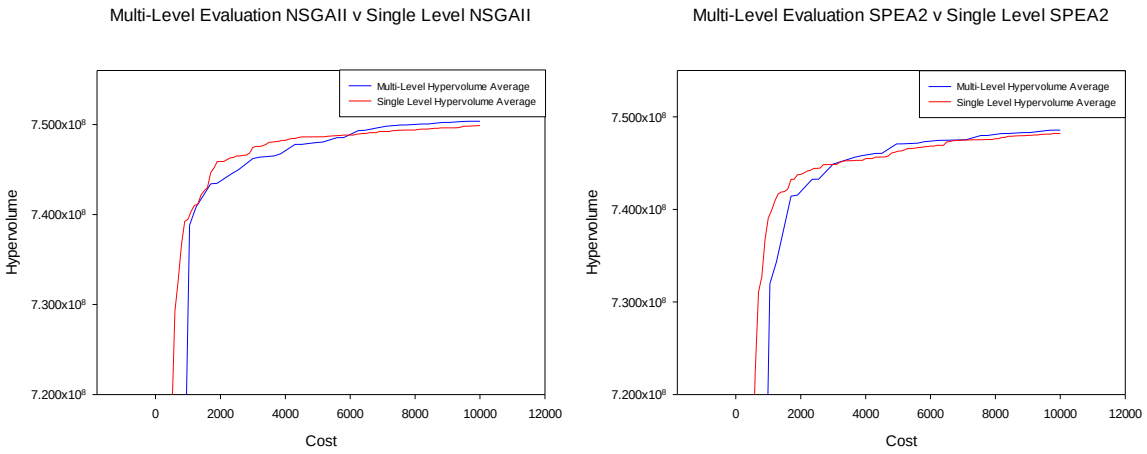


Figure 7.13 System - Device level average hypervolume results for the 5 runs of the multi-level evaluation and single level NSGAII and SPEA2 strategies * (s^U s^M s^L) [182000, 4150]

The scale at which each level contributes to the overall performance is hard to quantify. The system multi-level approach saw performance from lower levels migrate upwards allowing solutions evolved at lower levels to contribute to the default highest level population. Figure 7.14 tracks the values for the filter frequency error objective of the best solution found at each level over the design process for a number of runs. In both run 1 and run 2 NSGAII examples

solutions with improved filter frequency error are passed upwards to the level 3 population. There is however a slight sense of caution with this assessment as the analysis performed at level 2 is different to that at level 3 which perform as the default level of analysis. Level 2 only samples a third of the filter frequency, however this occurs at a much smaller frequency window which can mitigate some of this sampling error as values outside the 7.5 kHz / 12.5 kHz range are often below the -20dB threshold and therefore do not contribute any error as a result. Also the transfer of solutions from one level to another and the improvement in performance is relative, with a general lag in improvement at level 3 in comparison with the lower level indicating solutions migrating upwards. A look at the migration data and the success rate of solutions moving from level 3 to level 2 and also level 2 to level 3 as shown in figure 7.15 highlight the generally character of solutions migrating between the levels with those solutions from level 2 proving more successful in joining the higher level 3.

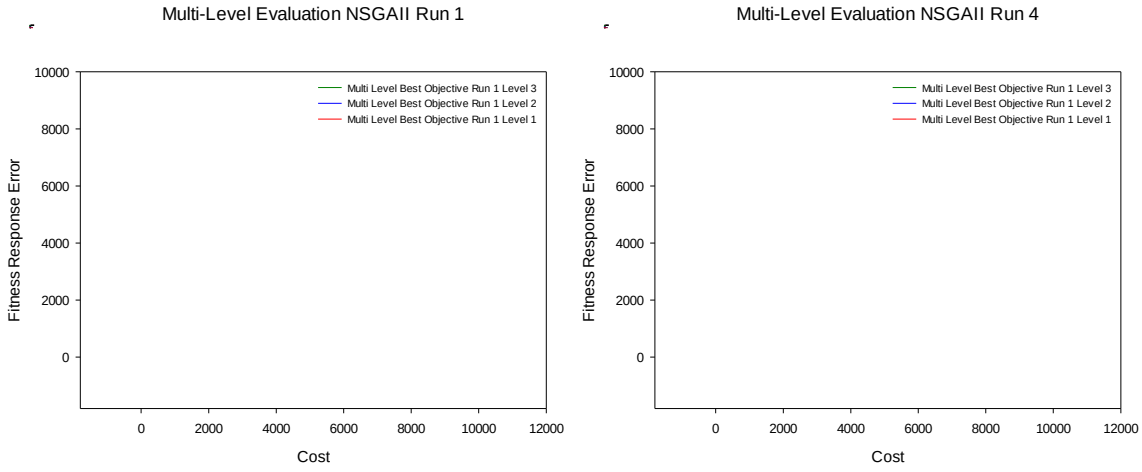


Figure 7.14 Best filter response objective values for multi-level evaluation NSGAII run 1 and 4 plots for levels 1, 2 and 3.

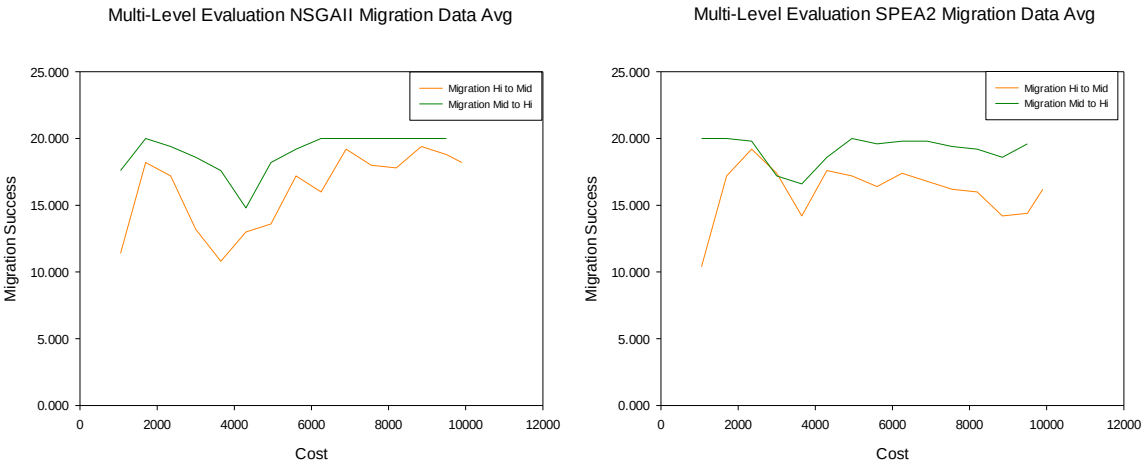


Figure 7.15 Generational average migration successes for multi-level evaluation NSGAII and SPEA2 strategies.

7.1.4 Multi-level parameterization

The approach of evolving a MEMS bandpass filter through direct synthesis of the 2D layout of the resonator components that make up the whole device brings with it an increase in the number of design variables which can influence the solutions phenotype. Similar to the system multi-level parameterization strategy, here two separate levels of representation are used to ease the pressure on the optimizer. Outlined in figure 7.17 the system – device multi-level parameterization design problem contains a default representation and a simplified representation which contains a number of global variables used to clone variables for any other resonator 'tanks'. The effect of cloning on each additional tank is shown in figure 7.16 with the majority of additional tank variables cloned.

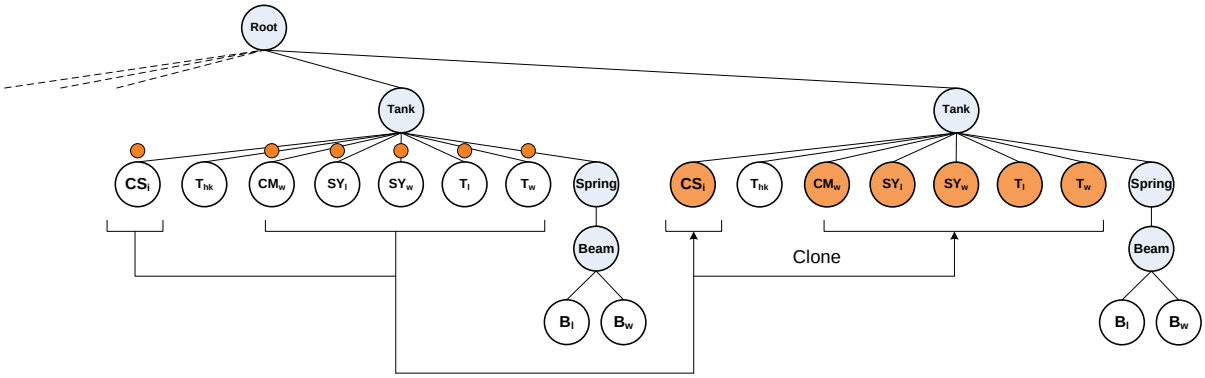


Figure 7.16 Multi-level parameterization global leaf variable clones

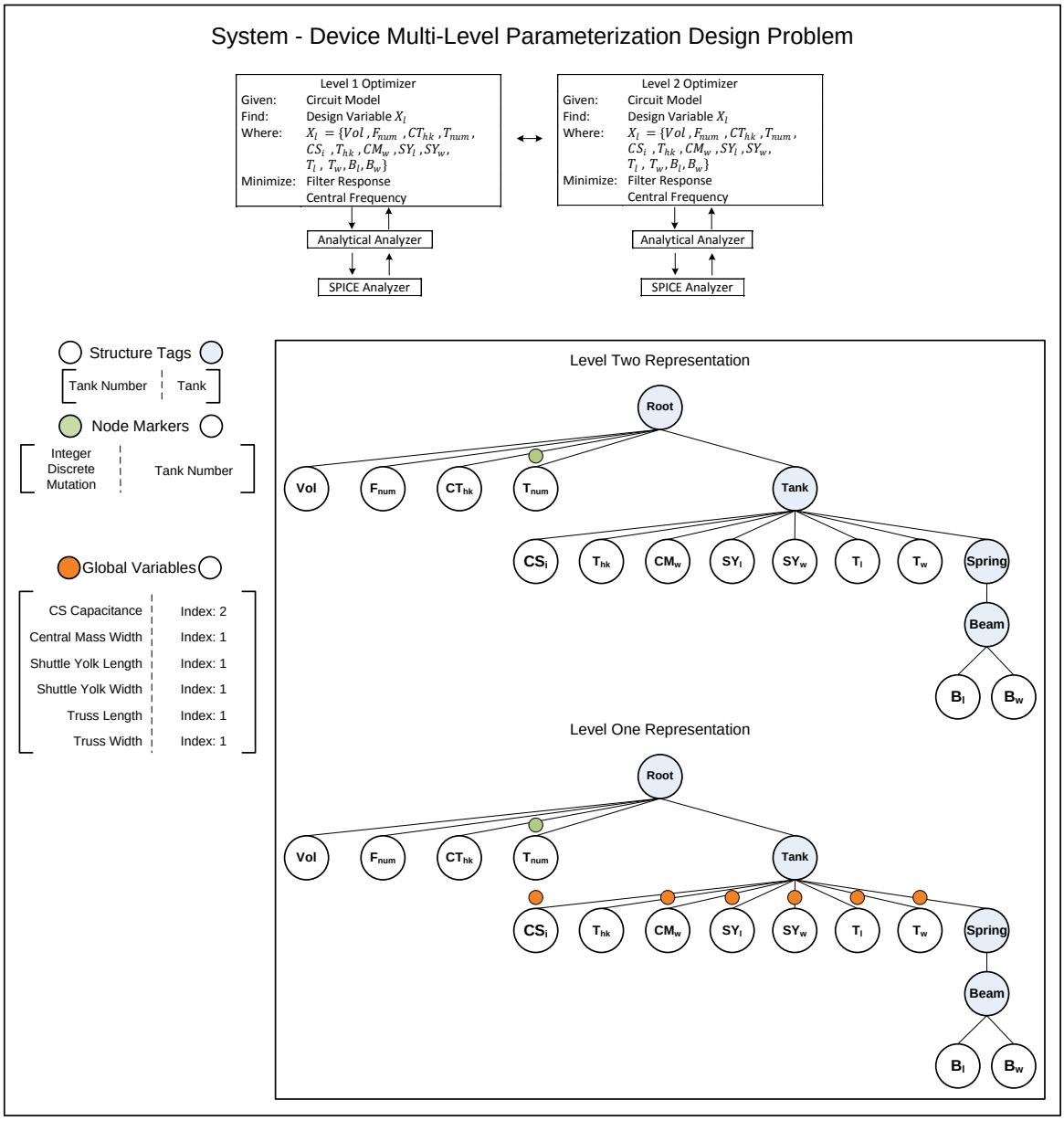


Figure 7.17 System - Device multi-level parameterization design template, with overview of problem, default representation, associated structure tags, node markers and global variables.

The choice of design variables which are to be global follow the approach used at the system level design problem representation, where the variables for inductance and coupling spring capacitance were used, here values associated with resonator mass and coupling spring capacitance are used. The effect on the number of design variables available for variation is shown in table 7.11 with the lowest level containing two thirds the number of variables of the higher level for a three tank filter and unable to reach variable counts for higher order filters that are open to level two representation. This is a result of a restriction on the upper bound for the tank number variable seen in table 7.12 and constitutes the only change to the design problem variable values from the default single level.

Table 7.11 System – Device Multi-Level Parameterization Level Variable Count

	Level One		Level Two		
	1 Tank	3 Tanks	1 Tank	3 Tanks	5 Tanks
Variable Count	13	19	13	31	49

Table 7.12 System - Device Multi-Level Parameterization Level One Filter Problem Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Tank Number	Integer	1	3

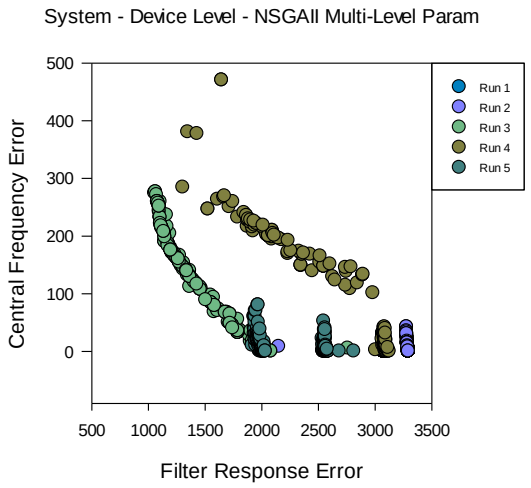
Table 7.13 Migrator Module Parameters for Multi-Level Parameterization

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	10	2
Level 2	Level 1	50	2

The migrator module for this design problem acts between the two levels, the parameters of which are shown in table 7.13, here the cycle count between each migration event has been reduced to allow more migration to occur, and the percentage increased for level 2 and reduced for level 1.

7.1.5 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 7.18 and fully in appendix D.1. Also shown are the best frequency transmissions for each algorithm, ranked by the filter frequency objective in figure 7.19 and fully in appendix D.1 along with their objective values in table 7.14. The mass and stiffness values for these solutions are shown in table 7.15 along with the phenotypes for the NSGAII strategy in appendix D.1. Finally the hypervolume values for both algorithms are shown in table 7.15, with the best results shaded.



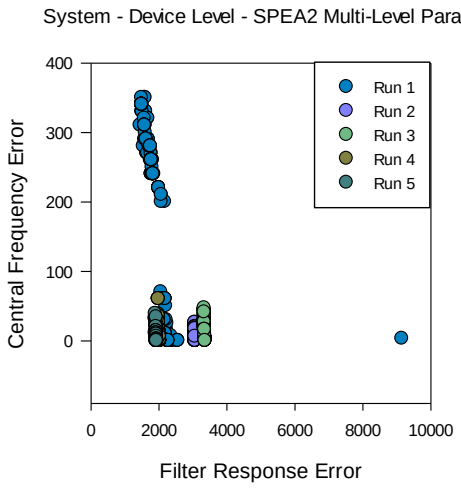


Figure 7.18 System – Device level run 1 – 5 final population sets for (left) NSGAII multi-level parameterization and (right) SPEA2 multi-level parameterization

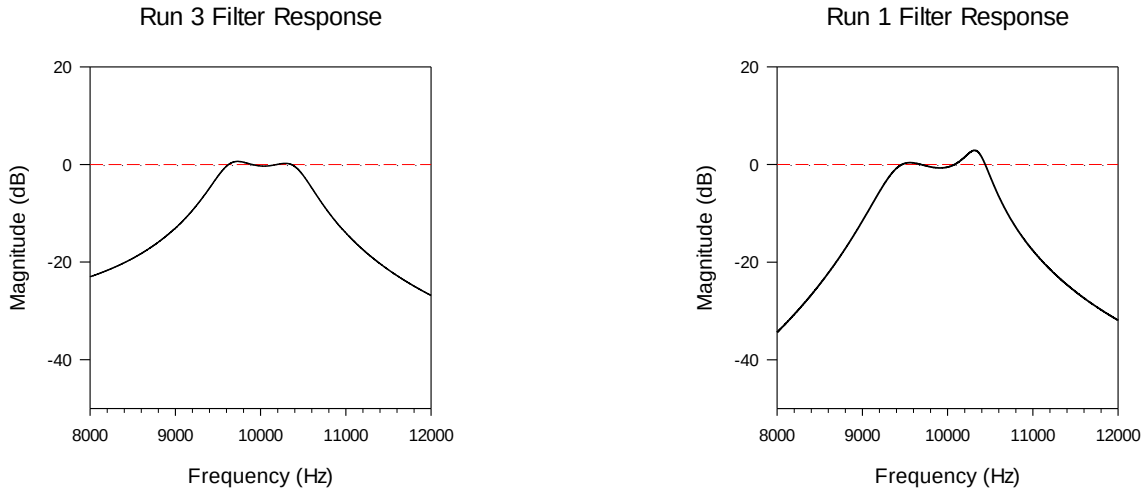


Figure 7.19 System – Device level best filter response ranked by filter frequency objective for (left) NSGAII multi-level parameterization and (right) SPEA2 multi-level parameterization

Table 7.14 System – Device Multi-Level Parameterization Bandpass Filter Results

System – Device Multi-Level Parameterization NSGAII						
Test	Index	Level	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	3	Low	3068.357	14	152.03	2
2	0	Hi	2152.948	8	131.01	9
3	51	Hi	1057.117	274	36.75	3
4	30	Hi	1305.991	284	71.59	3
5	3	Hi	1918.225	10	77.77	3
System – Device Multi-Level Parameterization SPEA2						
Test	Index	Level	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	20	Hi	1456.128	310	161.33	3
2	0	Hi	3064.001	20	134.70	2
3	23	Hi	3336.149	40	77.27	2
4	5	Low	1977.486	24	100.67	3
5	2	Hi	1888.774	32	81.66	3

Table 7.15 System – Device Multi-Level Parameterization Mechanical Values for Mass and Stiffness for the Best Solutions Ranked by the Filter Error Objective for Both NSGAII and SPEA2 Algorithms.

System – Device Multi-Level Parameterization NSGAII						
Individual Folded Flexure Resonator Values		Experimental Run				
		1	2	3	4	5
Tank 1	Mass (kg)	6.689E-08	6.683E-09	8.537E-08	7.717E-08	6.168E-08
	Stiffness (N/m)	266.772	23.766	365.375	334.027	236.354
Tank 2	Mass (kg)	2.732E-08	1.766E-09	5.552E-08	7.474E-08	6.060E-08
	Stiffness (N/m)	9.772	40.522	112.252	279.197	18.263
Tank 3	Mass (kg)	-	2.168E-08	8.324E-08	1.929E-08	6.799E-08
	Stiffness (N/m)	-	83.641	308.251	67.496	282.948
Tank 4	Mass (kg)	-	2.141E-08	-	-	-
	Stiffness (N/m)	-	80.607	-	-	-

Tank 5	Mass (kg)	-	2.462E-08	-	-	-
	Stiffness (N/m)	-	98.046	-	-	-
Tank 6	Mass (kg)	-	2.056E-08	-	-	-
	Stiffness (N/m)	-	69.585	-	-	-
Tank 7	Mass (kg)	-	1.734E-08	-	-	-
	Stiffness (N/m)	-	83.679	-	-	-
Tank 8	Mass (kg)	-	2.065E-08	-	-	-
	Stiffness (N/m)	-	83.679	-	-	-
Tank 9	Mass (kg)	-	2.142E-08	-	-	-
	Stiffness (N/m)	-	49.459	-	-	-

System – Device Multi-Level Parameterization SPEA2

Individual Folded Flexure Resonator Values		Experimental Run				
		1	2	3	4	5
Tank 1	Mass (kg)	8.120E-08	2.357E-07	3.759E-08	3.339E-08	9.507E-08
	Stiffness (N/m)	345.607	944.162	129.944	130.045	398.025
Tank 2	Mass (kg)	5.739E-08	7.594E-08	4.007E-09	5.034E-08	9.894E-08
	Stiffness (N/m)	212.712	24.593	39.220	90.430	91.270
Tank 3	Mass (kg)	3.003E-08	-	-	3.220E-8	9.069E-08
	Stiffness (N/m)	99.705	-	-	135.845	344.807

Table 7.16 System - Device Level Hypervolume Results for NSGAII and SPEA2 Multi-Level Parameterization

System – Device Multi-Level Parameterization		
Hypervolume	NSGAII	SPEA2
S^U	44616865.3595209	42348676.6911852
S^M	40364887.46608	38203098.87236
S^L	34658175.7364645	33318801.8980195

* (S^U S^M S^L) [10000, 5000]

The multi-level parameterization strategy ultimately performs the worst of all the strategies employed so far with only 3 out of 10 experimental runs providing solutions with suitable filter transmission shapes. The best solutions found ranked by filter frequency error in figure 7.19 and appendix figures D.13 and D.14 seem to fall into two distinct categories, those that provide suitable filter transmission shapes often similar to those found at the system level design problem, and finally those that consist of a single peak. Naturally the filter frequency error values reflect each of these phenotypes as seen in table 7.14 with subpar filter frequency error and central frequency values often within 40 Hz.

The hypervolume values show NSGAII to be once again superior to SPEA2 for this particular design problem, however still underperforming when compared with the single level strategy as seen in figure 7.20 with the generational hypervolume performance. Interestingly where one might expect the lowest level of parameterization to perform worse in terms of hypervolume value as seen for the NSGAII algorithm it is not the case for SPEA2, however this may simply be a result of early convergence of both population sets.

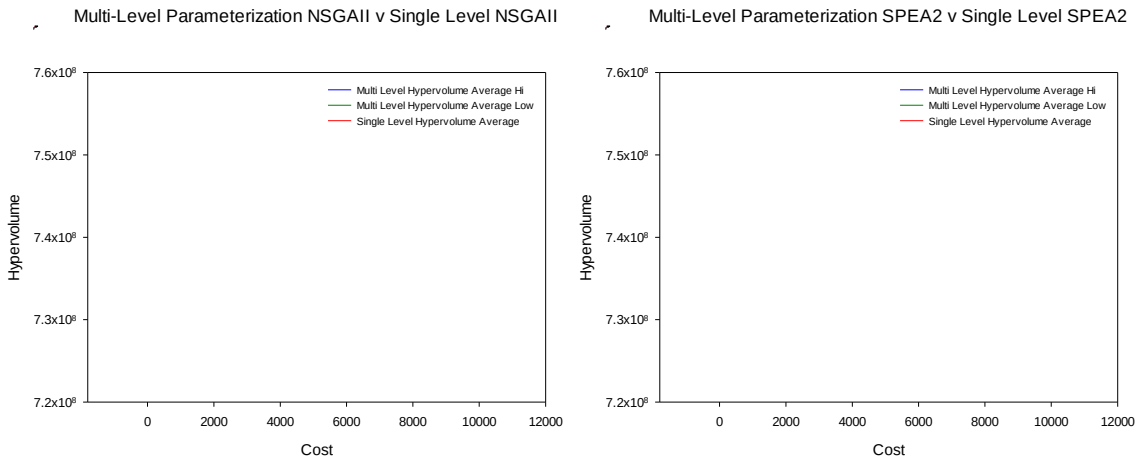


Figure 7.20 System - Device level average hypervolume results for the 5 runs of the multi-level parameterization and single level NSGAII and SPEA2 strategies * (S^U S^M S^L) [182000, 4150]

The disappointing performance may stem from a number of factors as a result of the multi-level strategy involved. A clear reason for the poor performance is the focus of the strategy to develop phenotypes which contain only a single peak and then have no ability to break away and evolve better frequency transmission solutions. Both level one and level two have the ability to evolve better filter frequency solutions over the design process as seen in figure 7.21. An example of three separate runs highlights the best filter frequency error objective value present in each population set from levels one and two over the design process.

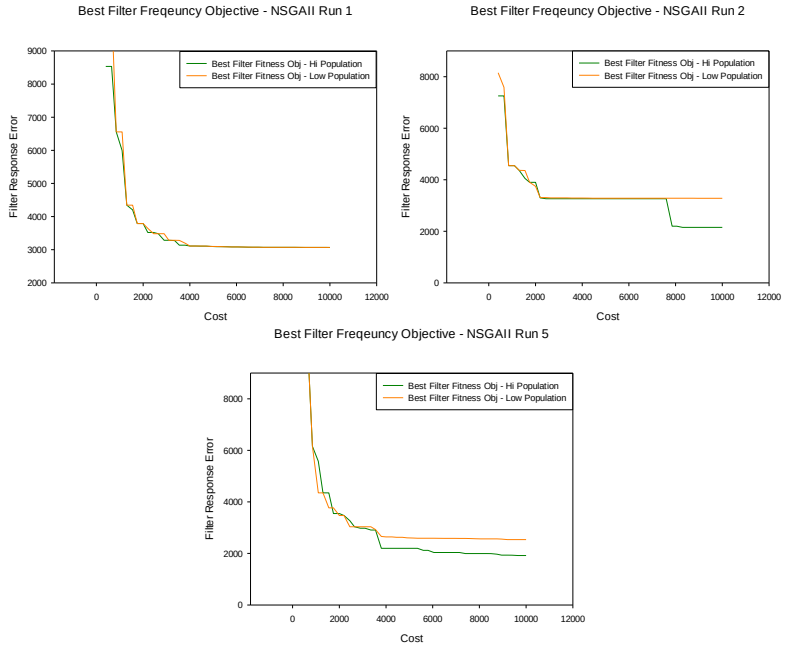


Figure 7.21 Best filter response objective values vs cost for multi-level parameterization NSGAII runs 1, 2, and 5 for levels one and two.

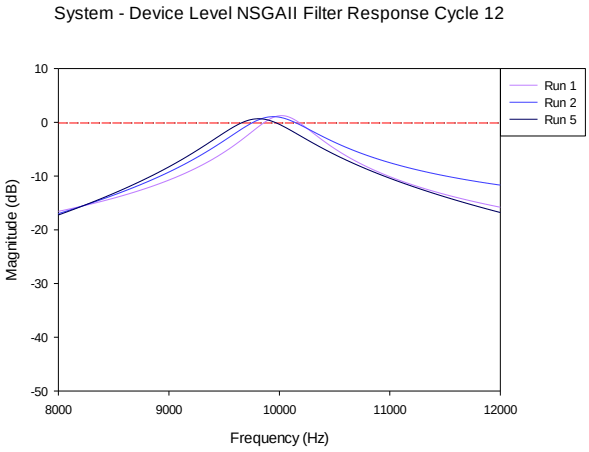


Figure 7.22 System - Device level best filter response ranked by filter frequency objective for NSGAII multi-level parameterization level one, runs 1, 2, and 5 at cycle 12

In all three examples the optimizer has evolved solutions which contain a single peak early on in the design process, as shown in figure 7.22 at cycle 12 or functional cost of around 3000. Early on in the design process a good number of improvements to the filter frequency error objective have come from the lowest level of parameterization. These solutions are migrated upwards and contribute to the higher level of parameterization. The downside however is that such phenotypes seem harder to improve upon for this particular strategy, whether a direct result of a simplified representation which can concentrate the population solutions to a particular genotype which it seems cannot be evolved sufficiently away from this filter transmission shape.

7.1.6 System – Device level comparison and analysis

The hierarchical nature of MEMS design modelling and synthesis provide the designer with a number of tools in which to go about building and optimising a device. Tools found at the system level provide a designer a number of abstract model representations through the use of block diagram, electrical equivalent or bond graph representations. Such tools and models allow large complex MEMS devices, made up of a number of components to be modelled and analysed at a reduced accuracy but also a reduced computational cost allowing automated design synthesis to occur more effectively.

Once a design has been found which matches the target function / performance for the MEMS device than focus can then turn to synthesizing the physical device itself and the 2D layout design. This can involve extracting information from the system level model and using this to define design target goals which will match the functional behaviour of the optimised system level model. However this may not always be possible, target behaviour may be unattainable as the design target goals are simply non-feasible and / or constraints at a device level arise which garner the same effect.

A separate approach which couples together the abstract modelling and analysis of the system level modelling and the realisation of the 2D layout modelling of the device level can possibly bridge this problem. A coupled system and device level bandpass filter design problem where the physical design of a bandpass filter layout is used and then modelled at the system level for analysis was undertaken successfully. This approach proved to be robust method with superior

performance in design and evidence to suggest that further optimisation would lead to greater performance when compared with the standard system level only approach in chapter 7. A list of the hypervolume performance for each of the three strategies employed for this particular design problem is shown in table 7.16. Both the single level and multi-level evaluation strategies stand out with both showing equal performance though on average the multi-level strategy performs better.

Table 7.17 System – Device Level Hypervolume Results for Single and Multi-Level Strategies for Both NSGAII and SPEA2 Algorithms.

System – Device Level NSGAII			
Hypervolume	Single Level	Multi-Level Evaluation	Multi-Level Parameterization
S^U	45293277.4844393	45231930.4653885	44616865.3595209
S^M	43500546.71823	44060613.34740	40364887.46608
S^L	42936836.5109377	40901705.6449109	34658175.7364645
System – Device Level SPEA2			
Hypervolume	Single Level	Multi-Level Evaluation	Multi-Level Parameterization
S^U	43783808.3056236	43629486.6022534	42348676.6911852
S^M	41489209.49112	41920157.36378	38203098.87236
S^L	38997795.1789861	40124595.2576051	33318801.8980195

* (S^U S^M S^L) [10000, 5000]

There are some interesting pieces of information to take from the set of multi-level results. Once again the multi-level evaluation strategy was able to boost performance whilst showing a positive linear trend for performance at the final stages of the design process. There are also it seems drawbacks to using some multi-level strategies, in this case multi-level parameterization when undertaken on certain kinds of design problems. In this case the convergence to certain phenotypes and the inability to evolve design away from this particular region of the design search space.

7.2 Physical – Device level design optimisation

Coupled design problems are a common theme within the both the MEMS and larger engineering industries as a whole [203][204]. MEMS can contain numerous components all functioning towards a final system goal, and it is their complex interaction and strong dependence on coordination that can make design synthesis in particular through automated methods difficult.

The previous system – device level design problem focused upon the ability for designers to utilize two or more levels for modelling and analysis to provide a platform for the design synthesis of a MEMS bandpass filter. The physical – device design problem takes a similar tract by combining the modelling of a device level folded flexure resonator component and a physical level coupling spring to create a coupled problem for MEMS design optimisation.

The physical – device design problems concerns itself with the physical coupling between the two components and the effect this has on optimisation and synthesis. The previous uni – level design problems for both the folded flexure resonator and coupling spring are incorporated with design objectives for the synthesis of both the folded flexure resonator and coupling spring.

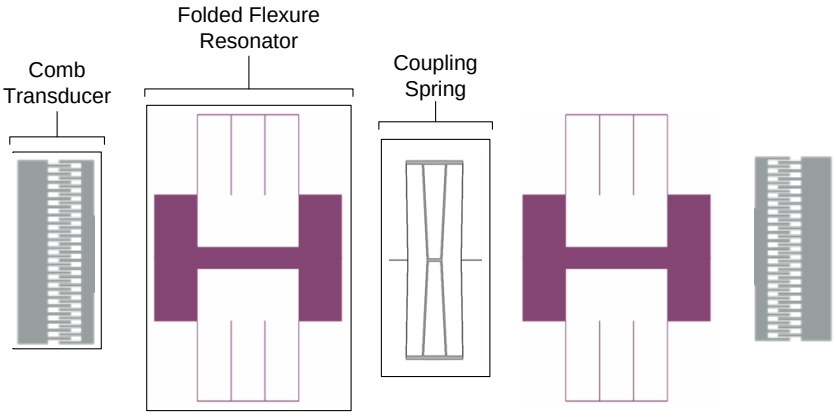


Figure 7.23 Physical level coupling spring and device level folded flexure resonator components

When looking to design a MEMS device or some of its components it may not be possible or practical to perform such synthesis in an isolated and individual manner. The interactions between such individual components may give rise to constraints which are simply not present or accounted for as separate design problems. The MEMS bandpass filter can be broken up into a number of components, three of which are shown in figure 7.23. As discussed in previous chapters the function of the MEMS filter can be described simply as the transfer of an input electrical force, conversion into mechanical energy and then filtered through a number of mechanical components before being reconverted into an electrical output signal. Here the force leads to the mechanical folded flexure resonator component to undergo transformation and displacement before passing on this force to the connected coupling spring component. The force exerted on the coupling spring and subsequently the displacement of the device leads to the accumulation of stresses upon the structure. The previous physical level design problem explored this particular phenomenon by casting it as a design objective to minimize the total von Mises stress within the coupling spring during displacement by an applied force.

The nodal folded flexure resonator model used in this design problem has been upgraded to include a comb transducer for evolving the force applied to the coupled coupling spring. The problem variables are shown in table 7.18 and consist of the variables for both the Sugar folded flexure resonator and coupling spring models used in the previous device and physical level problems along with the new comb transducer variables for force calculation.

The objective and constraints used in the physical – device design problem are shown in table 7.19 and are also an accumulation of those within the previous device and physical level problems, with the synthesis objectives for stiffness kx and mass error of the folded flexure resonator combined with the stiffness kx error of the coupling spring problem into a single mechanical error weighted sum objective as shown in equation 7.1. The constraints used are those associated with each model from both the device and physical level design problems.

The synthesis objective target information for both the folded flexure resonator and coupling spring are shown in table 7.20 and remain unchanged from the previous design problems.

Table 7.18 Physical - Device Level Filter Variable Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Central Mass Length (μm)	Real Valued	10	600
Central Mass Width (μm)	Real Valued	10	400
Shuttle Yolk Length (μm)	Real Valued	2	400
Shuttle Yolk Width (μm)	Real Valued	10	400
Beam Number	Integer	1	6
Beam Angle	Real Valued	45	135
Beam Length (μm)	Real Valued	5	100
Beam Width (μm)	Real Valued	2	20
Truss Width (μm)	Real Valued	2	20
Anchor Placement Length (μm)	Real Valued	0	400
Thickness (μm)	Real Valued	2	200
Truss Beam One Length (μm)	Real Valued	5	100
Truss Beam One Width (μm)	Real Valued	1	10
Truss Beam One Length (μm)	Real Valued	5	200
Truss Beam One Width (μm)	Real Valued	1	10
Mass Connector Length (μm)	Real Valued	2	100
Mass Connector Width (μm)	Real Valued	1	20
Centre Beam Length (μm)	Real Valued	20	100
Centre Beam Width (μm)	Real Valued	2	10
Outer Flexure Beam Width (μm)	Real Valued	1	10
Inner Flexure Beam Width (μm)	Real Valued	1	10
Flexure Length (μm)	Real Valued	50	400
Centre Gap Length (μm)	Real Valued	10	50
Coupling Spring Thickness (μm)	Real Valued	1	5
Voltage	Real Valued	1	200
Finger Number	Integer	1	200
Comb Thickness ($\boldsymbol{\mu m}$)	Real Valued	2	30

Table 7.19 Physical - Device Level Filter Problem Information

Objectives		Constraints	
Mechanical Error	Minimize	Constraint 1 [Truss Beam Two Length – (Inner Flexure Beam Width * 2)]	Inequality ≥ 0
von Mises Stress	Minimize	Constraint 2 [Centre Beam Length – ((Inner Flexure Beam Width * 2) + 3e-06)]	Inequality ≥ 0
		Constraint 3 [((Centre Beam Length / 2) + Centre Gap Length + Mass Connector Length) – ((Truss Beam Two Length / 2) + Truss Beam One Length + 10e-06)]	Inequality ≥ 0
		Constraint 4 [Shuttle Yolk Length - Central Mass Width]	Inequality ≥ 0
		Intersection Check	Inequality

By having to undertake both physical and device level analysis at the same time for a single level approach to design optimisation a designer is restricted to a lower number of functional evaluations as a result of the high cost for physical level design optimisation. The device level modelling and analysis in terms of computational cost is magnitudes of order smaller than that at the physical level analysis presented so far yet it is tied to the design synthesis of the physical level coupling spring.

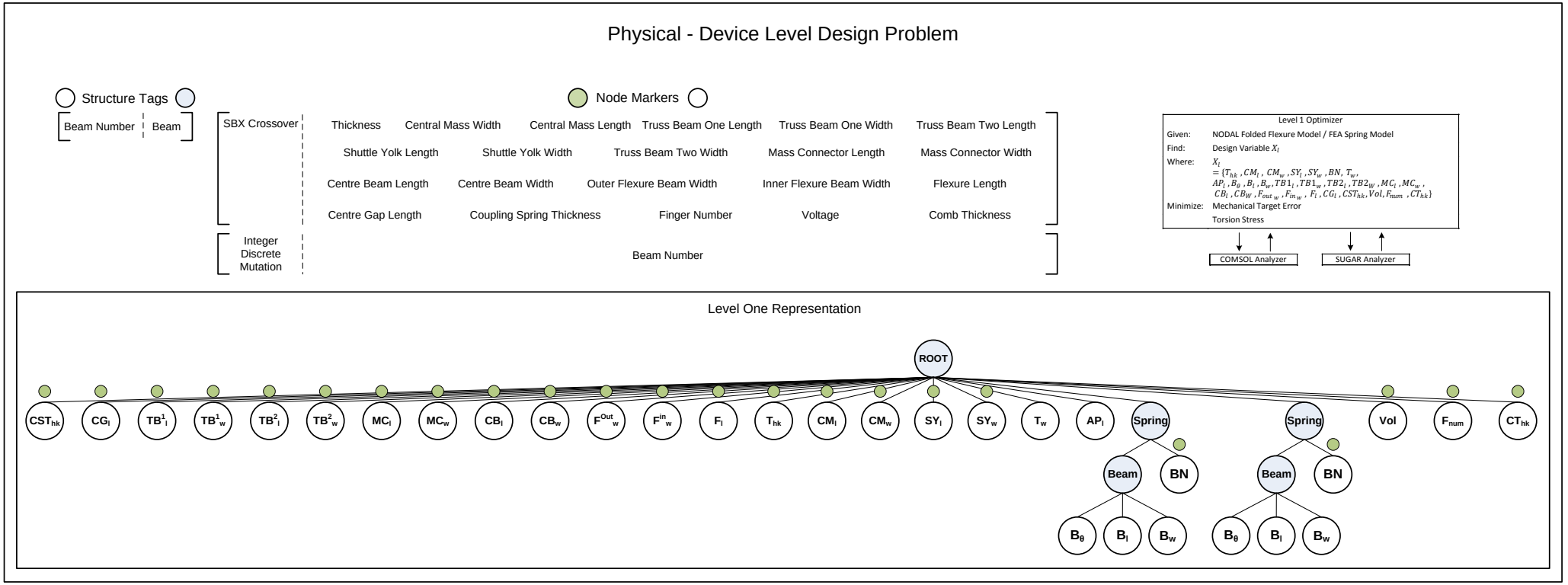


Figure 7.24 Physical – Device level design template, with overview of problem, default representation, associated structure tags and node markers.

Table 7.20 Physical – Device Level Synthesis Objective Target Information

Synthesis Objective	Target Value
Coupling Spring Stiffness Kx Error	21.55 N/m
Folded Flexure Stiffness Kx Error	2.45 N/m
Folded Flexure Mass Error	5.12E-10

Mechanical error:

$$\left[\left(\frac{1}{2.45}\right)\cdot\ |(Nodal\ Stiffness\ Kx\ Error)|\right] + \left[\left(\frac{1}{5.12e^{-10}}\right)\cdot\ |(Mass\ Error)|\right] + \left[\left(\frac{1}{21.55}\right)\cdot\ |(Comsol\ Stiffness\ Kx\ Error)|\right] \quad (7.1)$$

The overall physical – device level design problem is shown in figure 7.24 along with the structure and node markers associated with the updated representation. The design problem requires the construction of two new framework modules shown in figure 7.25 to hold both the problem and constraint information required for the design process. Analysis for both the NODAL folded flexure resonator and Comsol coupling spring models is done using the previous analysis modules built for the separate physical and device design problems, with an update to allow force calculation using local design variables rather than a fixed value.

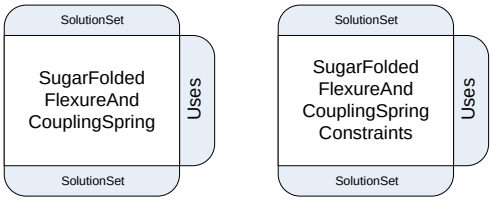


Figure 7.25 Physical – Device level modules for (a) sugar folded flexure and coupling spring problem information and (b) sugar folded flexure and coupling spring constraints.

Algorithmic parameters for both NSGAII and SPEA2 remain unchanged from those used in the previous default uni level design problems, with the overall design process incorporating both device level design and coupling spring design the inclusion of a structural crossover module for the folded flexure springs is added. The overall number of functional evaluations and the population sizes shown in table 7.21 reflect the previous physical level design problem due to the dominating cost of coupling spring analysis. As a result there is a reduction in the available folded flexure resonator analysis when compared with the previous device level problem.

Table 7.21 Physical - Device Level Algorithmic Parameter Changes

Algorithm Parameter	Default Value
Population Size	50
Offspring Size	50
Selection Size	50
Replacement Size	100
Generations	20

7.2.1 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 7.22 and fully in appendix D.2. A list of the best results ranked by mechanical error is shown in table 7.3 and the phenotypes for the NSGAII strategy are shown in figure 7.3. Finally the hypervolume values for both algorithms are shown in table 7.23, with the best results shaded.

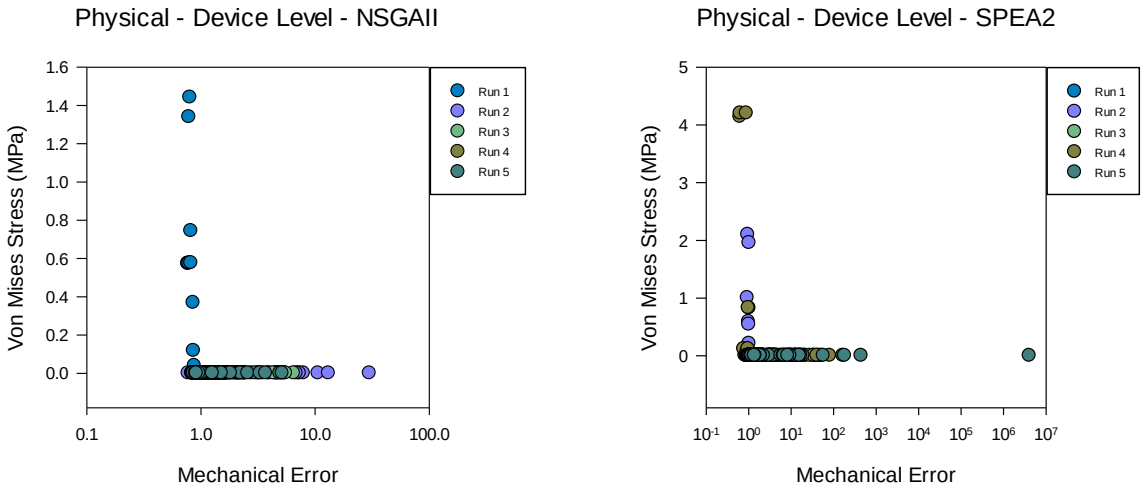


Figure 7.26 Physical – Device level run 1 – 5 final population sets for (left) NSGAII and (right) SPEA2

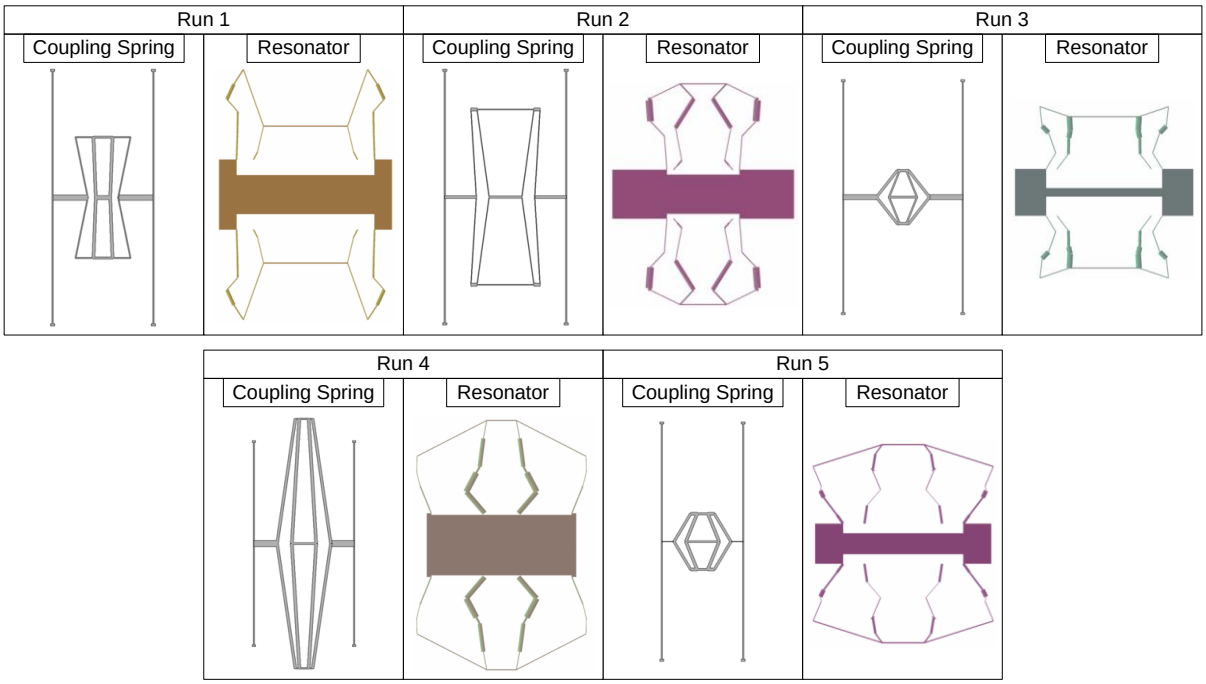


Figure 7.27 Physical - Device Level run 1 – 5 coupling spring and folded flexure best result ranked by mechanical error objective for NSGAII

Table 7.22 Physical – Device Level Best Results Ranked by Mechanical Error

Physical – Device Level NSGAII				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Mechanical Error	von Mises Stress (MPa)
1	9	0	0.77355695	0.5721191
2	4	0	0.77834212	3.04789853E-05
3	4	0	1.1470202057	9.40711458E-07
4	7	0	0.875476104	0.0008675434
5	8	0	0.8354898155	1.12458886E-06
Physical – Device Level SPEA2				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Mechanical Error	von Mises Stress (MPa)
1	13	0	0.9085316797	5.91594286E-05
2	7	0	0.8159659583	2.05859201E-05
3	11	0	0.989889265	0.00014922
4	8	0	0.625449884	4.13938734
5	9	0	0.861200768	7.4977586E-06

Table 7.23 Physical - Device Level Hypervolume Results for NSGAII and SPEA2

Physical – Device Level		
Hypervolume	NSGAII	SPEA2
S^U	338.087859942512	349.351035125636
S^M	325.1325906	331.3561860
S^L	304.048705831891	320.808824069444

* (S^U S^M S^L) [5, 80]

The design synthesis and optimisation of two main components within a MEMS bandpass filter was the objective for the physical – device level design problem. Coupled together into a single design problem the task of synthesizing both the folded flexure resonator and coupling spring to match a number of design targets using both the NSGAII and SPEA2 algorithms is shown to be unsuccessful. Both algorithms for this single level strategy failed to synthesis the combined components within range of a desirable design target of 1%.

The population sets for both algorithms show parity to evolve solutions which hit a wall of optimization towards the mechanical error objective at around 1.0 total error equivalent to 100% total deviation from the synthesis target goals set out in table 7.20. The ability to control the force applied to the coupling spring component and therefore more efficiently minimize the total von Mises stress of the component provides both solutions with very low stress levels, especially in comparison to those found at the physical level design problem, but also much larger values as a result of force values higher than $1e^{-6}$ N.

The phenotypes for both of the evolved folded flexure resonators and coupling spring components for NSGAII best solutions ranked by mechanical error are shown in figure 7.27 with a noticeable mix in the shape and sizing of the coupling spring components with longer and more flexible or smaller and compact designs. This mix is similar to those evolved at the uni level physical design problem with an association of small compact structures to low levels of von Mises stress comparatively to the much larger and flexible designs. This characteristic of the phenotypes is also portrayed within the objective values in table 7.22.

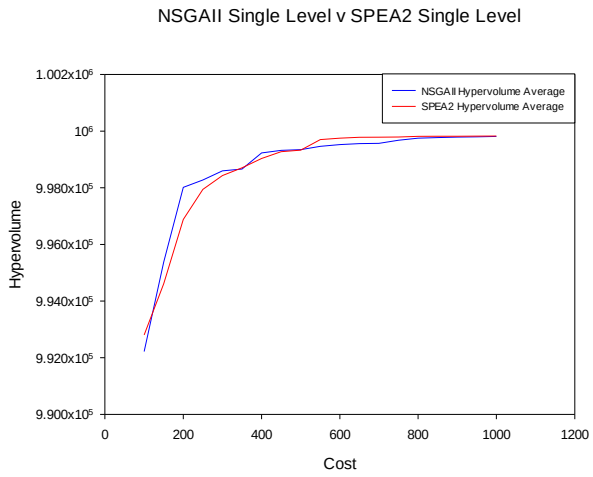


Figure 7.28 Physical - Device level average hypervolume results for the 5 runs of the single level NSGAII and SPEA2 strategies * ($S^U S^M S^L$) [5000, 200]

Looking at the generational plots for the hypervolume values for both the NSGAII and SPEA2 algorithms show similar performance throughout the design process which is reflected in the individual final hypervolume results in table 7.23. The choice of the nadir point for hypervolume calculation as shown here effects the scale of difference between each algorithms performance, with SPEA2 seemingly performing better in the final solutions found at each run and the final hypervolume results compared with NSGAII than what is portrayed in figure 7.28.

Taking into account all results would lead to the conclusion that both the high error in NSGAII run 3 and lower error in SPEA2 run 4 are outliers and that both algorithms have similar performance overall.

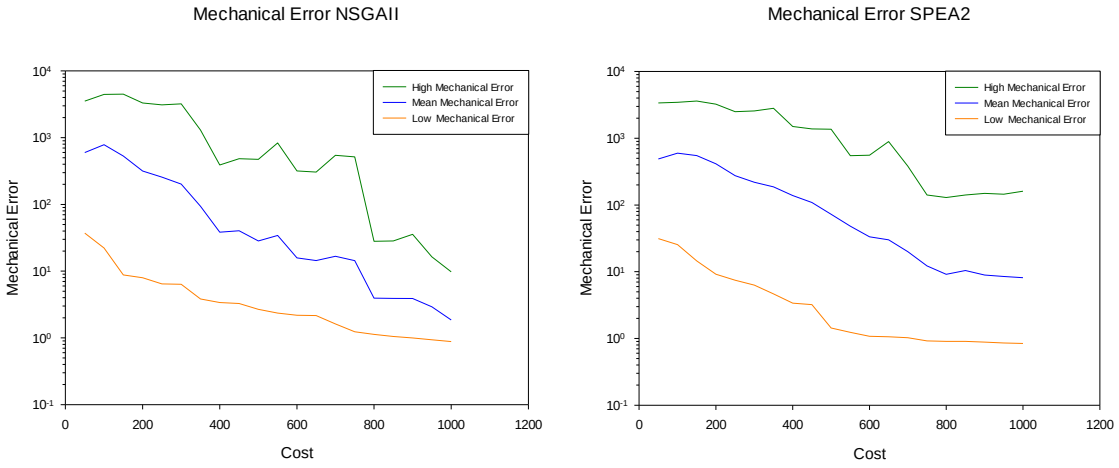


Figure 7.29 Average high, low and mean mechanical error for (left) NSGAII and (right) SPEA2

Expanding on the results and looking in depth at what factors lead to improvement within both the objectives set out for this design problem. To begin with a look at the average mechanical error for both algorithms over the entire design process in figure 7.29 show a strong trend towards improving the total synthesis error for both the folded flexure resonator and the coupling spring. Both algorithms are able to produce solutions with a mechanical error of around 1.0, though with diminishing improvement over the final half of the design process. The mean

values also show continued improvement with a characteristic link to the performance of the solution with the highest error in the population, no doubt dragging down mean performance. The algorithm SPEA2 performs worst in its mean mechanical error as a result of the poor performance late on for the highest mechanical error when compared with NSGAII which shows a continual linear improvement.

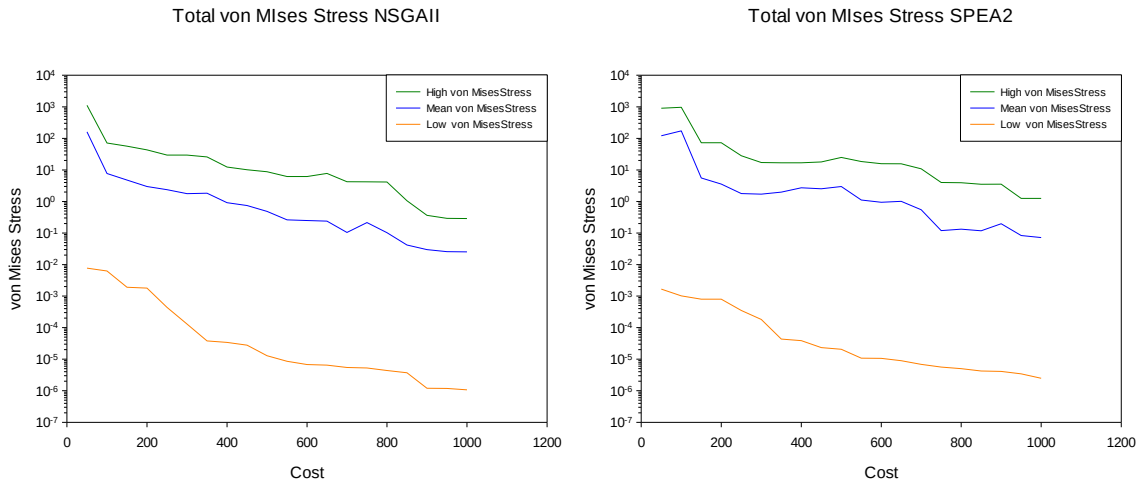


Figure 7.30 Average high, low and mean von Mises stress values for (left) NSGAII and (right) SPEA2

The objective for minimizing the total von Mises stress of the coupling spring component shown in figure 7.30 also shows continual improvement throughout the population set over the design process however the spread between high, low and mean values is more distorted with the average von Mises stress levels much closer to the higher levels of stress. The ability to lower the applied force on the coupling spring component and the natural evolution towards compact structures are the contributing factors to improved stress values when compared with those evolved under the uni level physical design problem.

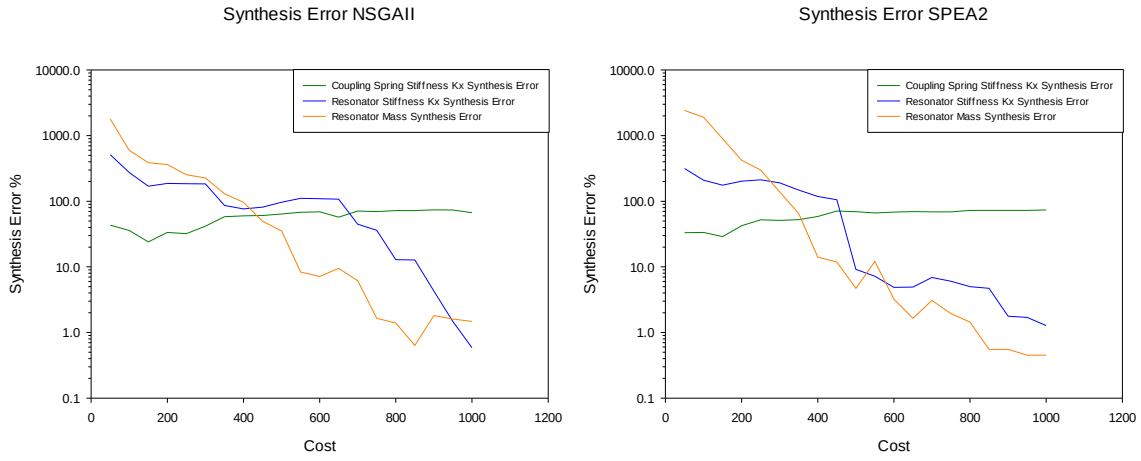


Figure 7.31 Average best synthesis % values for folded flexure stiffness kx, mass and coupling spring stiffness kx targets for (left) NSGAII and (right) SPEA2.

Looking back over the results presented it can be seen that both the mechanical error and von Mises stress objectives can be optimized by both algorithms over the design process. However

the optimizers are unable to break through a mechanical error wall and evolve solutions which are significantly below 1.0 / 100% synthesis error. Figure 7.31 breaks up the contributing factors to the mechanical error objective, these being the folded flexure stiffness kx, mass and coupling spring stiffness kx error. It shows a stark picture on the performance of the individual algorithms to be unable to synthesize the coupling spring to match the design targets. Where both the folded flexure stiffness kx and mass targets are met within 1% error the coupling spring stiffness kx error actually gets worse over the design process going from around 30% error to above 80% error.

The encapsulation of the various synthesis objectives for both the folded flexure resonator and the coupling spring into a single objective has led to the inability of the optimizer to synthesize one of the components which previously it was able to do so as a single uni level design problem. One possible reason is the inability to distinguish the coupling spring stiffness kx error from the mechanical error objective and with selection pressure as a result of the other von Mises stress objective driving designs of a compact structure; solutions which retained some level of flexibility nearer the stiffness target goals are lost.

The folded flexure resonator synthesis targets seem unaffected and surprisingly given the small number of functional evaluations the optimizer is able to evolve solutions near the target goal of 1% synthesis error. The comb transducer design variables for the update folded flexure resonator affect the von Mises stress objective values for the solution but do not directly affect the folded flexure resonator in either of its synthesis objectives allowing the design to evolve freely.

7.2.2 Multi-level evaluation

The ability to increase the number of functional evaluations over the design process is a goal of any designer looking to implement some form of automated design synthesis. Expensive modelling and analysis techniques such as FEA found within the physical level of MEMS design process can lead to a restriction in the number of functional evaluations spent searching for design solutions to the design problems set out by the designer.

A number of hierarchical levels available to a MEMS designer have been explored in chapter 6 through the construction of specific design problems tailored to each level. The application of a multi-level evaluation strategy has also been applied to solve each of these design problems using only tools and models available to that specific level.

The application of a multi-level strategy which spans two levels of the hierarchical modelling process of MEMS is undertaken here to solve the coupled physical – device problem. The previous physical level multi-level evaluation strategy used a number of FEA models of varying computational cost and accuracy in order to increase the number of available functional evaluations. Here a device level NODAL model is constructed using the same design parameters as the FEA model used previously and allow transition of each model from one level to another easier as no conversion is needed. This is not the case however with the folded flexure component which uses both the analytical model and the NODAL model built from the Sugar platform.

The benefit of using more than one level of the design process to model a device or component is that it often brings with it a larger computational cost saving than would possibly be had if such a strategy was used only at a single modelling and analysis level. The cost ratio between the expensive physical level FEA model and the inexpensive device level NODAL model are shown in table 7.24 with the device level model providing analysis for 100th of the cost of the physical level model. This compares with only providing half the cost reduction when using a similar strategy only with physical level tools as seen in the previous chapter.

Table 7.24 Evaluation cost for NODAL and FEA coupling spring model

NODAL Analysis	FEA Analysis
Mean Analysis Time (20 Calls)	Mean Analysis Time (20 Calls)
0.0548502105	6.0
Ratio	
1:109.388824	

There is however a trade-off in the accuracy of the evaluation between the device and physical level models. Figure 7.32 plots the displacement analysis for both NODAL and FEA models of the coupling spring over a number of different input force values and two separate FEA models of varying meshing granularity. The difference between the NODAL and default FEA coupling spring models are considerably larger than the two FEA models which utilize different meshing coarseness.

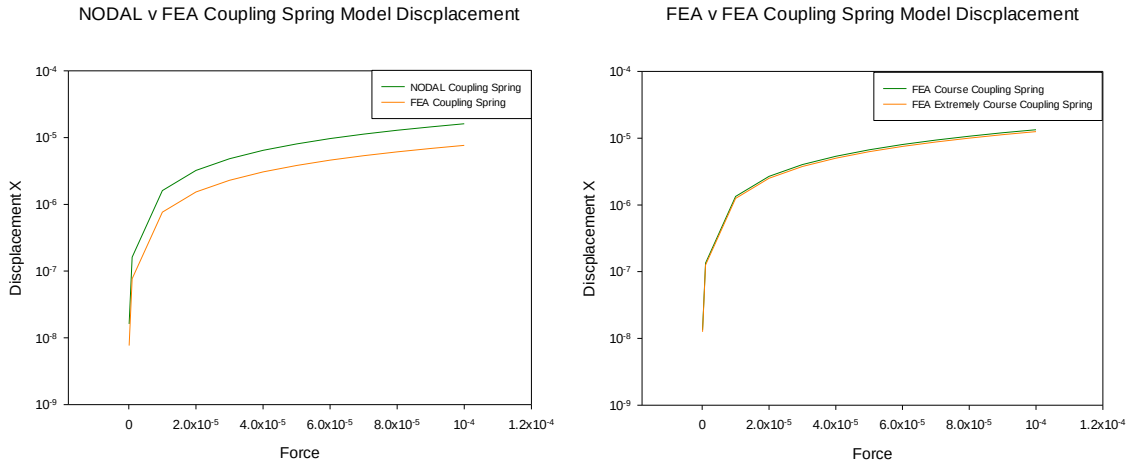


Figure 7.32 Coupling spring force vs displacement plots for (left) NODAL v FEA models and (right) FEA ‘course’ and FEA ‘extremely course’ models.

The multi-level evaluation design process is split into two levels, the first containing only device level modelling and analysis tools with the use of an analytical folded flexure resonator model and a NODAL coupling spring model to represent the entire coupled problem. Analysis of each component within this particular representation is done through the previous folded flexure resonator analysis module and an additional module for NODAL coupling spring analysis. A module for the lower level problem information has also been constructed to build the required representation, objective and constraint information and are all shown in figure 7.33.

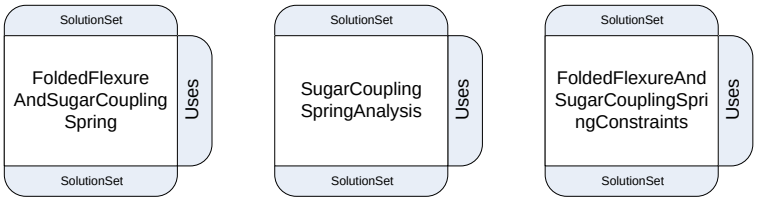


Figure 7.33 Physical - Device level modules for folded flexure and sugar coupling spring problem information (a) sugar coupling spring analysis (b) and folded flexure and sugar coupling spring constraints (c).

The transition from one level to another follows similar paths to the previous device multi-level evaluation strategy with the addition of a subtree converter module to the design process to allow conversion of the analytical folded flexure resonator representation into the NODAL resonator representation used at the default level. Overall two migrator modules are used to allow individuals to move between the neighbouring levels and the values for migration percentage along with the cycle count when migration is invoked are shown in table 7.25.

The overall cost of the design process remains at 1000 functional evaluations with each functional evaluation at the lowest level costing just a fraction at 0.01 default level functional cost. The lowest level pathway is called sequentially throughout the design process while the higher more costly level is called every ten cycles. This provides around 900 functional evaluations at the default higher level and 10,000 functional evaluations at the lowest level with a slight deviation given the need to analyse solutions migrating between levels

Table 7.25 Migrator Module Parameters for Multi-Level Evaluation

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	20	10
Level 2	Level 1	10	10

Figure 7.34 outlines the overall multi-level evaluation strategy for solving the coupled physical-device design problem, and includes both low and high level representations and their associated variables, structure tags and node markers. The addition of the lower level representation means a change in the objectives set out to solve as the device level tools can no longer undertake von Mises stress analysis. Therefore the mechanical objective has been split into two separate objectives with one focusing on the folded flexure resonator and the other on the coupling spring as shown in table 7.26. Additional constraints tied to the analytical folded flexure resonator model are also included while new low level analytical model design variables are held in table 7.27.

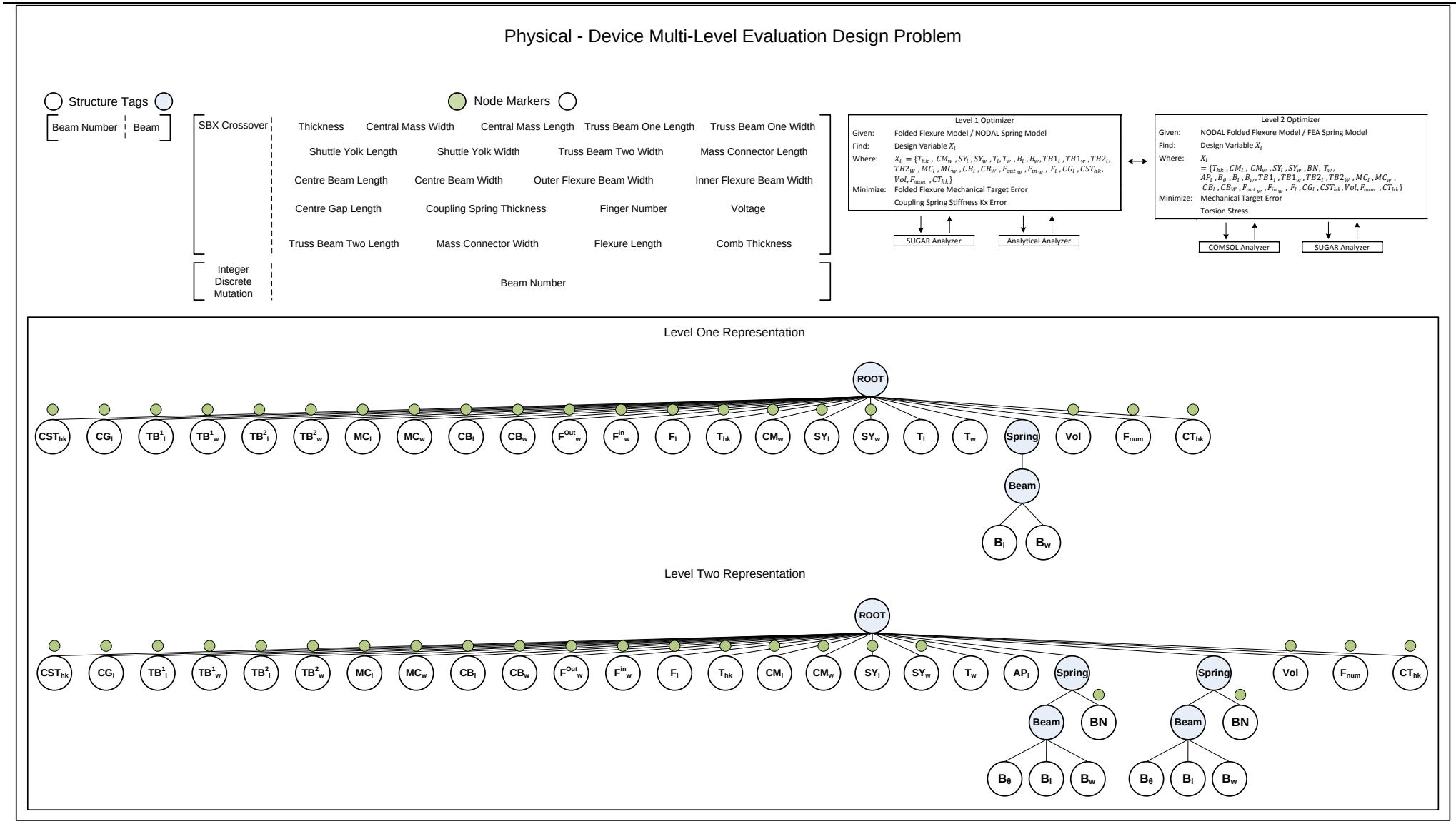


Figure 7.34 Physical – Device multi-level evaluation design template, with overview of problem, default representation, associated structure tags and node markers

Table 7.26 Physical – Device Multi-Level Evaluation Level One Problem Information

Objectives		Constraints	
Folded Flexure Mechanical Error	Minimize	Constraint 1 [Truss Beam Two Length – (Inner Flexure Beam Width * 2)]	Inequality ≥ 0
Coupling Spring Stiffness Kx Error	Minimize	Constraint 2 [Centre Beam Length – ((Inner Flexure Beam Width * 2) + 3e-06)]	Inequality ≥ 0
		Constraint 3 [((Centre Beam Length / 2) + Centre Gap Length + Mass Connector Length) – ((Truss Beam Two Length / 2) + Truss Beam One Length + 10e-06)]	Inequality ≥ 0
		Constraint 4 [Shuttle Yolk Length - Central Mass Width]	Inequality ≥ 0
		Constraint 5 (([Beam Width] * 4) - ([Truss Length] * 3)) + 9e-6	Inequality ≤ 0
		Intersection Check	Inequality

Table 7.27 Physical – Device Multi-Level Evaluation Level One Folded Flexure Parameters

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Central Mass Width (μm)	Real Valued	10	400
Shuttle Yolk Length (μm)	Real Valued	2	400
Shuttle Yolk Width (μm)	Real Valued	10	400
Beam Length (μm)	Real Valued	5	100
Beam Width (μm)	Real Valued	2	20
Truss Length (μm)	Real Valued	0	400
Truss Width (μm)	Real Valued	2	20
Thickness ($\boldsymbol{\mu m}$)	Real Valued	2	200

7.2.3 Numerical results

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 7.35 and fully in appendix D.2. A list of the best results ranked by mechanical error is shown in table 7.28 and the phenotypes for the NSGAII strategy are shown in figure 7.36. Finally the hypervolume values for both algorithms are shown in table 7.29, with the best results shaded.

Table 7.28 Physical – Device Multi-Level Evaluation Best Results Ranked by Mechanical Error

Physical – Device Multi-Level Evaluation NSGAII				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Mechanical Error	von Mises Stress (MPa)
1	9	0	0.072571025	0.07449602
2	14	0	0.088241182	65.291398
3	15	0	0.1039934666	42.715399
4	8	0	0.0443308767	0.00278349
5	10	0	0.23005965	0.0035231
Physical – Device Multi-Level Evaluation SPEA2				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Mechanical Error	von Mises Stress (MPa)
1	12	0	0.2895132	0.009786932
2	11	0	0.18666324	12.13550066
3	12	0	0.16812338	0.0432885703
4	7	0	0.14325546	2.2040436E-05
5	4	0	0.08002408	0.0003154865

Table 7.29 Physical - Device Level Hypervolume Results for NSGAII and SPEA2 Multi-Level Evaluation

Physical – Device Multi-Level Evaluation		
Hypervolume	NSGAII	SPEA2
S^U	396.452937469177	393.597901525275
S^M	387.3681500	385.1080075
S^L	380.342463184353	376.836837582899

* (S^U S^M S^L) [5, 80]

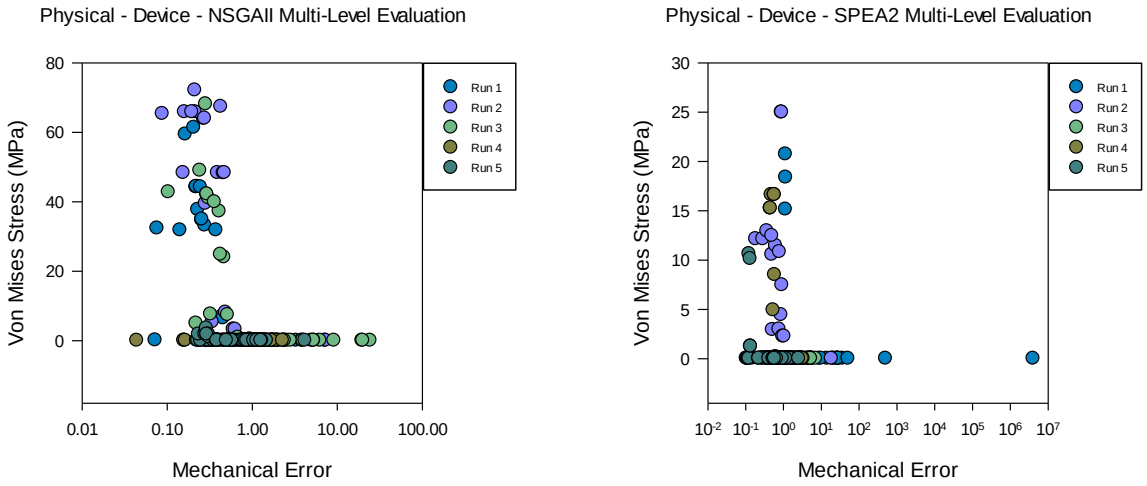


Figure 7.35 Physical – Device level run 1 – 5 final population sets for (left) NSGAII multi-level evaluation and (right) SPEA2 multi-level evaluation

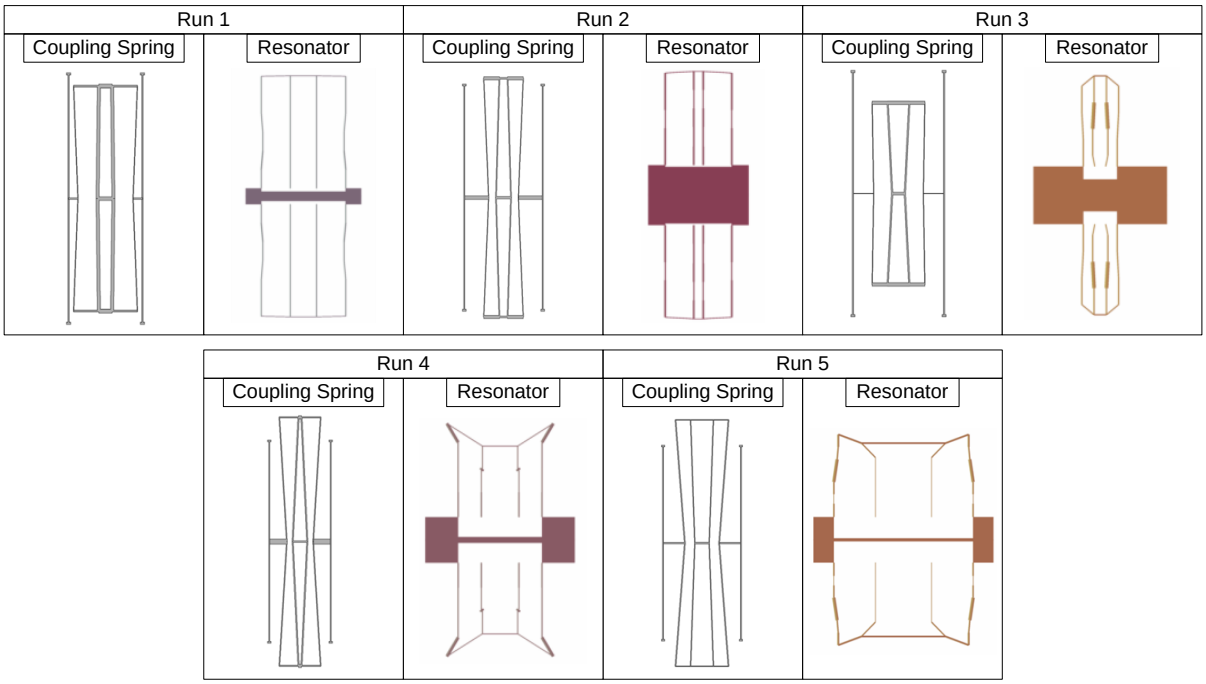


Figure 7.36 Physical - Device level run 1 – 5 coupling spring and folded flexure best result ranked by mechanical error objective for NSGAII multi-level evaluation

The ability to use a number of modelling and analysis tools and incorporate them into the design process of a MEMS device can be beneficial as it opens up the possibility of an increase in functional evaluations, can alter the search space as a result of different model representations and allow the use of simple or complex analysis to be undertaken.

The coupled physical – device level design problem looks to evolve the components of a MEMS bandpass filter in particular the folded flexure resonator and coupling spring components. The goals of such synthesis were to minimize the mechanical error for the synthesis targets of each component while also trying to minimize the von Mises stress levels of the coupling spring. The application of a multi-level evaluation strategy allows for additional levels of modelling to be incorporated, in particular the construction of a NODAL coupling spring model representation for simple and quick analysis of the device. The disadvantage is the loss of stress analysis however through connection to a much higher level FEA model this is still able to occur throughout the design process. This allows simple analysis to be undertaken with a reduction in the computational cost though with a loss of analysis fidelity, which is however made up using the more complex FEA model. The reduction in computational cost significantly increases the number of functional evaluations made available to the optimizer for a number of key synthesis objectives.

The application of the multi-level evaluation strategy has lead to a significant improvement in the overall performance of each optimizer in particular with regards to the mechanical error objective. The characteristics of the population sets as seen in figure 7.35 remain similar to the single level strategy with the exception of the ability to break through the mechanical error value 1.0 wall and produce solutions which match more closely the target synthesis goals set out. The phenotypes for the best solutions found ranked by mechanical error are shown in figure 7.36 for NSGAII and their objective values in table 7.28. These phenotypes highlight one possible contribution to the overall improvement in solution design, that being an increase in the number of flexible coupling spring shapes.

A breakdown of the mechanical error objective values into the constituent synthesis error parts for the best results ranked by mechanical error achieved by the single and multi-level evaluation strategy for NSGAII are shown in table 7.30. A direct comparison with the single and multi-level strategy highlights clearly the main difference in performance for this particular objective with the single levels main mechanical error contribution coming from high coupling spring synthesis error percentages. This is no longer the case with the multi-level evaluation strategy results where values of around 80% synthesis error for the single level strategy are reduced to an error of just around 5% on average. Coupled with the very low synthesis error values for the folded flexure resonator provides solutions which are superior in performance towards matching the design synthesis targets set out.

Table 7.30 Physical – Device Level and Multi-Level Evaluation Synthesis Error % Values for Best Results NSGAII Ranked by Mechanical Error

Physical – Device Level NSGAII				Physical – Device Multi-Level Evaluation NSGAII		
Test	Folded Flexure Stiffness Kx Error %	Folded Flexure Mass Error %	Coupling Spring Stiffness Error %	Folded Flexure Stiffness Kx Error %	Folded Flexure Mass Error %	Coupling Spring Stiffness Error %
1	1.8457	0.6691	74.841	3.5498	2.6444	1.0629
2	25.925	28.656	23.254	3.6243	2.9372	2.2626
3	21.547	12.778	80.377	2.9883	1.9103	5.5008
4	0.5909	7.0756	79.881	1.7841	2.6091	0.0399
5	0.8006	1.7040	81.044	5.0660	7.4339	10.506

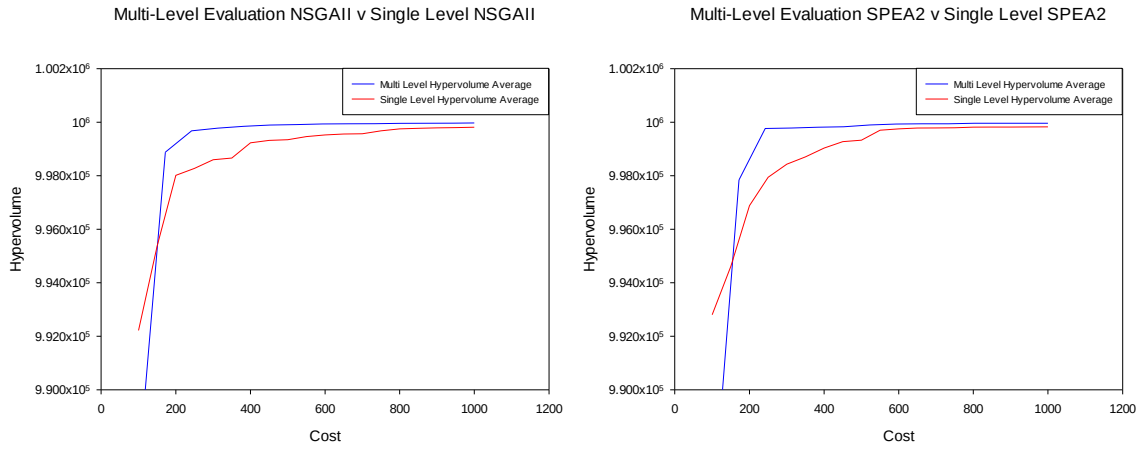


Figure 7.37 Physical - Device level average hypervolume results for the 5 runs of the multi-level evaluation and single level NSGAII and SPEA2 strategies * (s^U s^M s^L) [5000, 200]

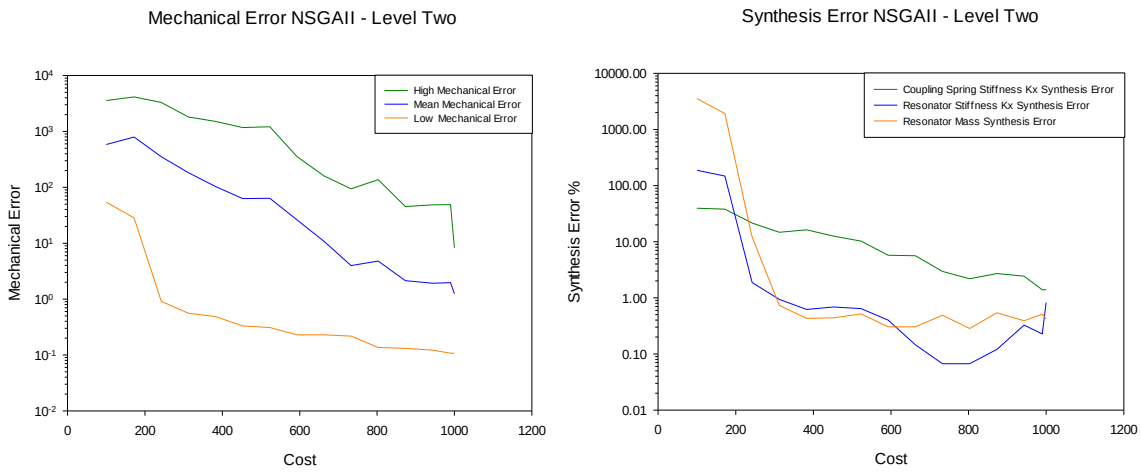


Figure 7.38 Average high, low and mean mechanical error for NSGAII (left) and average best synthesis % values for folded flexure stiffness kx, mass and coupling spring stiffness kx targets for NSGAII (right) multi-level evaluation strategy.

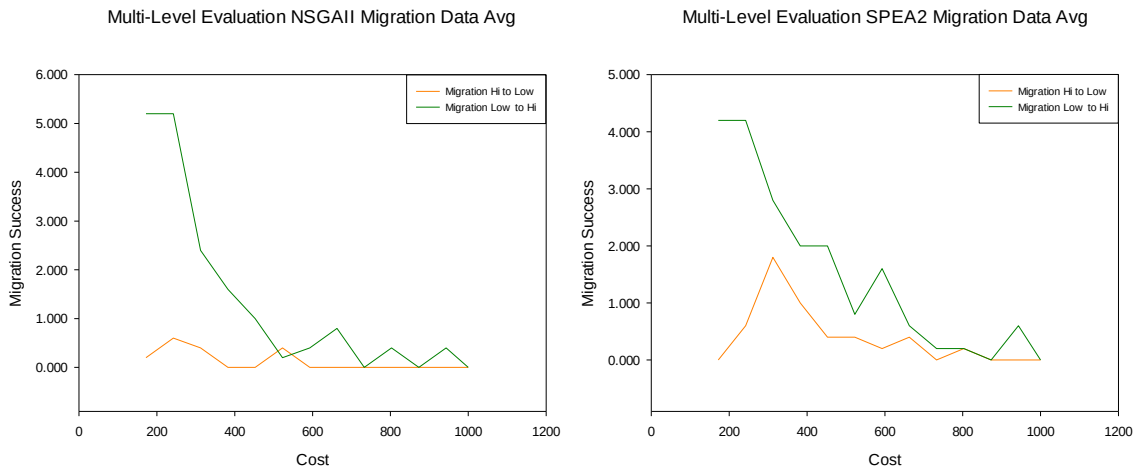


Figure 7.39 Generational average migration success for multi-level evaluation NSGAII and SPEA2 strategies.

The overall performance for each optimizer has also improved over the design process as shown with the generational hypervolume performance in figure 7.37 providing superior mean values at the end of the design process and a marked speed up in improvement early on, reaching hypervolume values at a fifth of the cost of the single level strategy.

The characteristic improvement of the mechanical error over the design process is also similar to that of the previous single level design strategy as seen in figure 7.38, with improvements to both high and mean values and a significant improvement in the lowest mechanical error solution within the population. The most noticeable change is the increased pace of improvement to the lowest mechanical error solution, reaching target values of around 0.1 mechanical error or 10% deviation at around a cost of 200 functional evaluations. Decomposing the mechanical error values into each synthesis error value as shown in figure 7.38 over the design process it can be seen that the synthesis targets associated with the folded flexure resonator are optimized very quickly with the coupling spring synthesis error itself progressing at a slower but linear pace. The characteristic change in coupling spring optimisation from the previous static or loss of improvement seen in the single level strategy to a more gradual creep towards optimisation has no doubt provided good solutions.

In what bearing has the additional level of optimisation lead to an improvement to the overall design process is not entirely clear. The use of analytical folded flexure resonator and NODAL coupling spring models allows more functional evaluations to be used throughout the design process which can possibly account for the quick improvement for the mechanical error objective as it is the individual synthesis targets that are optimised at this level only.

These optimised solutions then need to be migrated to the higher level and contribute positively to the population in order to remain. However a look at the migration data for both algorithms in figure 7.39 shows that this is not always the case with a very low number of solutions successfully migrating from one population to another in particular from level two to one. There are explanations for why this may be, particularly linked to the increased performance of solutions being found very quickly these solutions are passed on to the highest level and are retained and make it harder later on for other solutions to supplant them. The effect of transition from one model representation to another can also change the functional

characteristics of the components within each solution enough that they are no longer optimal, at least not locally to the level the solutions are migrating to. Those solutions which are retained can be evolved locally with an improvement in their mechanical error objectives and also their von Mises stress objective values putting more pressure on solutions looking to migrate upwards to level two. Looking back over the best solutions found and their phenotypes what is clear is that some form of local optimisation has occurred given the structure and shape of the folded flexure resonators no longer match those of the analytical models but the more adaptable and variable NODAL models of level two.

The application of the multi-level evaluation strategy leads to a boost in early optimisation particularly for the synthesis objectives of the folded flexure resonator and coupling spring components. Afterwards local improvement seems to occur more readily with level two adapting these good solutions provided by the lowest level and improving upon them.

7.2.4 Multidisciplinary optimisation

Microelectromechanical systems often consist of multiple components or devices which cooperatively interact to form some kind of function to meet some application or design goal they are targeted towards. Each of these devices or components can be associated with a single or multiple number of unique disciplines, be they electrical as found in the sensing transducer component of the bandpass filter or mechanical for the folded flexure resonator that makes up the filter itself.

Such multidisciplinary structures can be hard to synthesize and optimise as often for each increase in discipline brings with it the need to have knowledge within the field it operates. The application of multidisciplinary optimisation to such problems is an attempt to simplify and improve the overall design optimisation process through decomposition of such problems into smaller more discipline specific tasks.

The difficulty with MEMS and in particular the components that are contained within them is that they are often not decomposable structurally in such a way that they can be optimised separately without any need for communication of information between the decomposed objects. A clear example is shown in the device level design problem with the attempt to perform multidisciplinary optimisation on a single folded flexure resonator. Here a task to synthesis a resonator to match certain functional or physical attributes was undertaken, in this instance to match target values for the stiffness and mass of the device. The key components and design variables strongly associated with each of these attributes were chosen and decomposed from the entire design problem representation, with both the folded flexure spring component and central mass component contributing towards each objective.

Separate optimisation can then be allowed to occur for each component; however the entire folded flexure resonator model is not truly decomposable. In order to analyse each of the folded flexure spring or central mass components for optimisation of their local objectives, in this case to match the target values set out requires that the decomposed model be reconstructed. Otherwise after a number of iterations the evolved components in their isolation when combined eventually at the system level will more likely be constrained with folded flexure springs unable to combine to their separate evolved central mass components.

The physical – device level design problem offers a new challenge to the multidisciplinary optimisation strategy by containing two individual components that are fully separable in terms of the objectives and design variables set out within the problem. Each of the folded flexure resonator and coupling spring components can be optimised separately without the need to pass any genetic information and the design objectives and constraints themselves can also be decomposed into specific subsystems.

With the folded flexure resonator and coupling spring components decomposed into separated sections of optimisation a decision has to be made on whether to perform further decomposition of the individual components or simply evolve the whole components and objectives as one. Such decomposition has already occurred in the previous uni level device and physical design problem examples and with synthesis targets once again present for the folded flexure resonator and coupling spring such a strategy could also be beneficial. Therefore a further decomposition is undertaken which is similar to previous examples with the folded flexure resonator components broken down into subsystems for the folded flexure spring and central mass units, and likewise the coupling spring broken down into separate subsystems focusing on the synthesis and von Mises objectives.

The overall multidisciplinary design process is shown in figure 7.40 and contains the system and subsystem layout and the individual representations within them, for brevity some of the design variables in the subsystems are culled but still remain in each subsystem fully. Also included are the node markers used to differentiate design variables that are active within each subsystem and associated structure tags. Similar to the previous uni level examples SBX crossover only occurs within subsystems one, two and three and is ignored for the folded flexure spring variables, structural crossover still occurs for the folded flexure spring elements within subsystem four and the beam number variable is restricted to integer discrete mutation.

In order to undertake multidisciplinary optimisation for this coupled design problem requires the construction of additional objectives and constraints. Table 7.31 holds the objectives and table 7.32 the constraints active for the system and subsystem components of the design process with each focusing on a specific task. The system level remains unchanged from the default physical – device level problem, while subsystem one and two focus upon the coupling spring component and subsystem three and four upon the folded flexure resonator.

Table 7.31 Physical – Device Multidisciplinary Optimisation Objectives

	System Level		Subsystem 1	Subsystem 2		Subsystem 3	Subsystem 4
	Objective 1	Objective 2	Objective 1	Objective 1	Objective 2	Objective 1	Objective 1
Objective Type	Minimize	Minimize	Minimize	Minimize	Minimize	Minimize	Minimize
Objective Description	Stiffness Kx Error	Von Mises Stress	Coupling Spring Stiffness Kx Error	von Mises Stress	Displacement in X Axis	Folded Flexure Mass Error	Folded Flexure Stiffness Kx Error

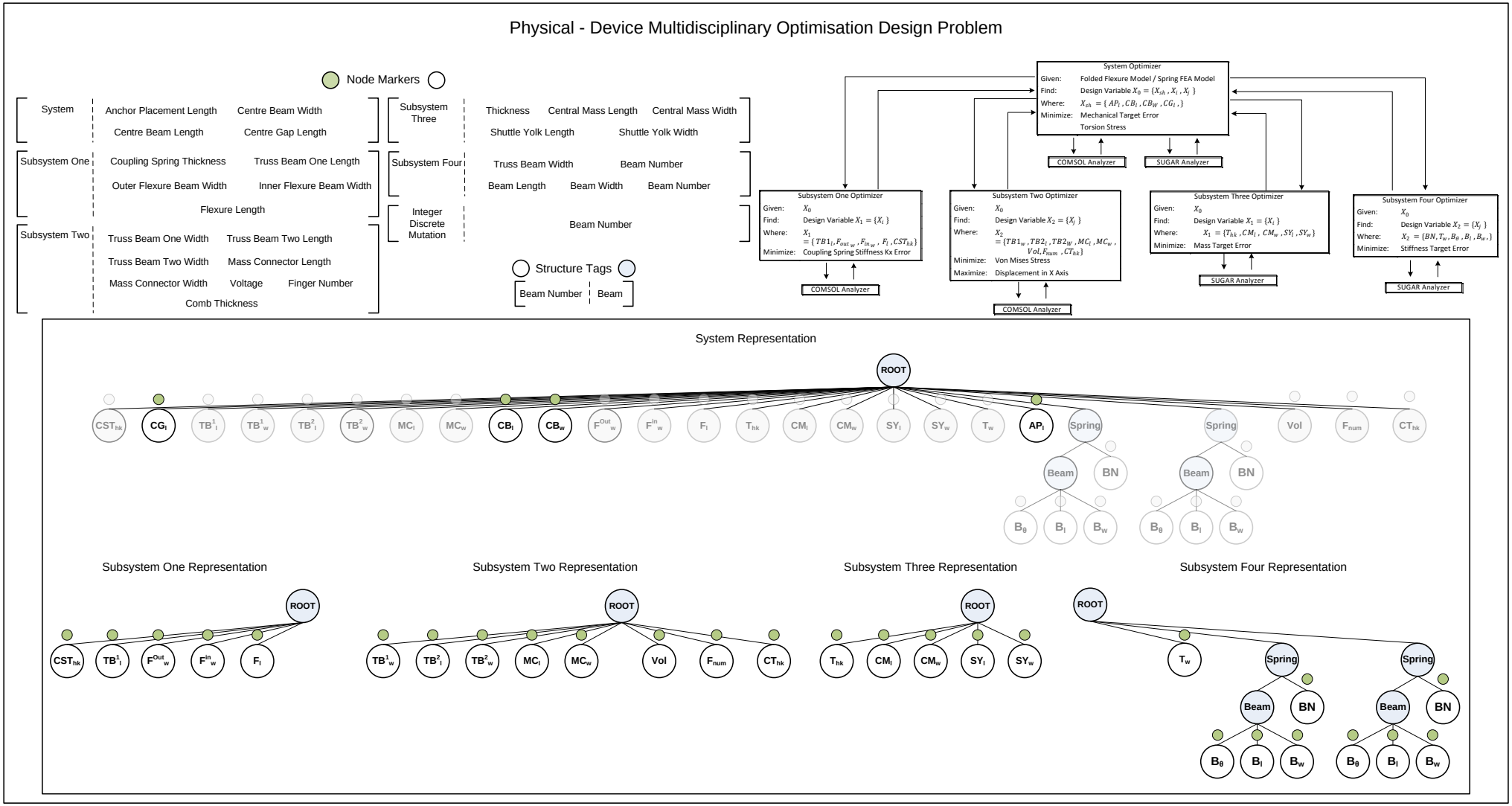


Figure 7.40 Physical – Device multidisciplinary optimisation design template, with overview of problem, default representation, associated structure tags and node markers.

Table 7.32 Physical – Device Multidisciplinary Optimisation Constraints

	Subsystem 1 / Subsystem 2			Subsystem 3 / Subsystem 4	
	Constraint 1	Constraint 2	Constraint 3	Constraint 4	Intersection
Constraint Type	Inequality ≥ 0	Inequality ≥ 0	Inequality ≥ 0	Inequality ≥ 0	Inequality
Constraint Description	[Truss Beam Two Length – (Inner Flexure Beam Width * 2)]	[Centre Beam Length – ((Inner Flexure Beam Width * 2) + 3e-06)]	[((Centre Beam Length / 2) + Centre Gap Length + Mass Connector Length) – ((Truss Beam Two Length / 2) + Truss Beam One Length + 10e-06)]	[Shuttle Yolk Length - Central Mass Width]	Intersection Check

The algorithmic parameters remain the same for both NSGAII and SPEA2 however the application of the multidisciplinary optimisation strategy requires a change in the population sets used within each subsystem in part due to the low number of functional evaluations available as a result of the computational cost from FEA analysis. Table 7.33 holds the population set values for the design process with a reduction to the system level offspring size, subsystem population set sizes that are also reduced to allow more generations to be undertaken over the whole design process.

Table 7.33 Physical – Device Level Multidisciplinary Algorithmic Parameter Changes

Algorithm Parameter	Default Value
System Population Size	50
System Offspring Size	25
System Selection Size	25
System Replacement Size	75
Grand Pareto Size	150
Subsystem Population Size	25
Subsystem Offspring Size	25
Subsystem Selection Size	25
Subsystem Replacement Size	50
Subsystem Total Size	100

The computational cost of the functional evaluations remains at 1000 with the system level pathway called every 50 generations, subsystems one and two called every 10 generations and finally subsystem three and four called sequentially throughout. This comes as a result of the reduced computational cost from analysing the folded flexure resonator model separately as seen within the previous multi-level evaluation strategy. The effect of the partition of the design problem into a number of system and subsystem pathways and the analysis undertaken within them gives rise to around 300 to 350 functional evaluations called for the system and subsystem one and two pathways and around 50 functional evaluations (5000 analysis calls) for both subsystem three and four.

The separation of the folded flexure resonator and the coupling spring components into a number of subsystems allows for both to undertake optimization in isolation from each other structurally, where there is no need to share genetic information between them. However subsystem two looks to evolve the coupling spring in order to minimize the von Mises Stress of the whole component and such analysis requires the calculation of the force applied to the spring as designated in the original design problem. Though the control of the comb transducer

variables are held within subsystem two the analysis is actually undertaken in a different subsystem, essentially two and three through the use of the analytical analyser. Therefore the comb transducer variables for each solution need to be passed to such subsystems for analysis and later have the results passed back to be held locally within subsystem two for its own analysis of the von Mises stress. Therefore in addition to a gene swapper module a new module coined an analysis swapper module shown in figure 7.41 is constructed to allow analysis values to be passed between solutions of each subsystem if necessary.

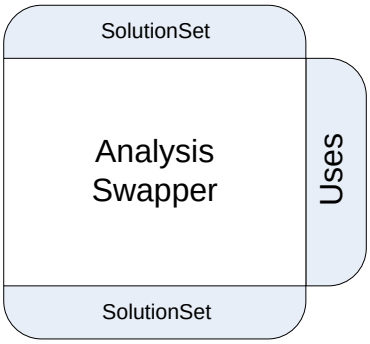


Figure 7.41 Physical - Device level module for analysis swapper

7.2.5 Numerical analysis

The results presented for both sets of experiments are the individual final population sets for each of the five tests performed by each algorithm, shown in figures 7.42 and fully in appendix D.2. A list of the best results ranked by mechanical error is shown in table 7.34 and the phenotypes for the NSGAII strategy are shown in figure 7.43. Finally the hypervolume values for both algorithms are shown in table 7.35, with the best results shaded.

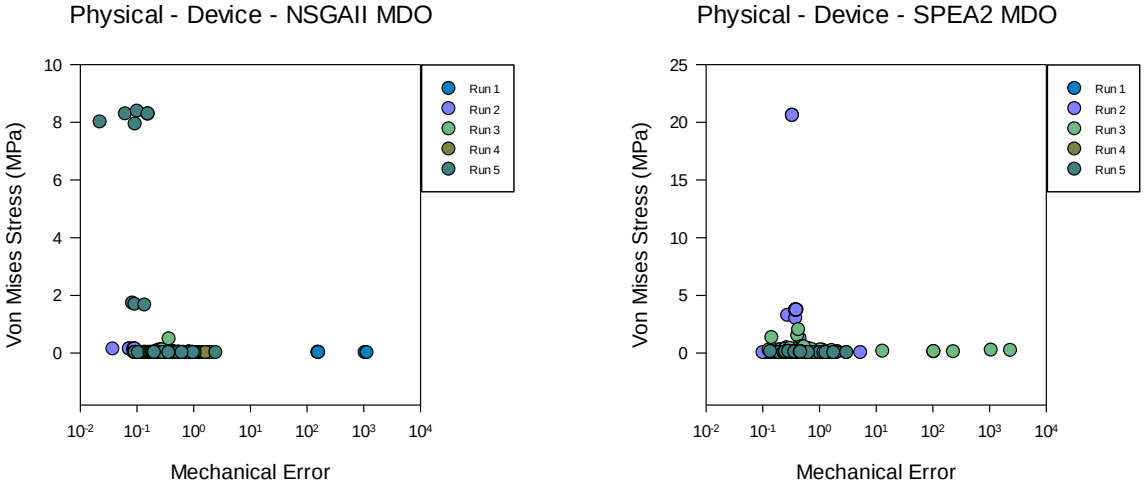


Figure 7.42 Physical – Device level run 1 – 5 final population sets for (left) NSGAII multidisciplinary optimisation and (right) SPEA2 multidisciplinary optimisation

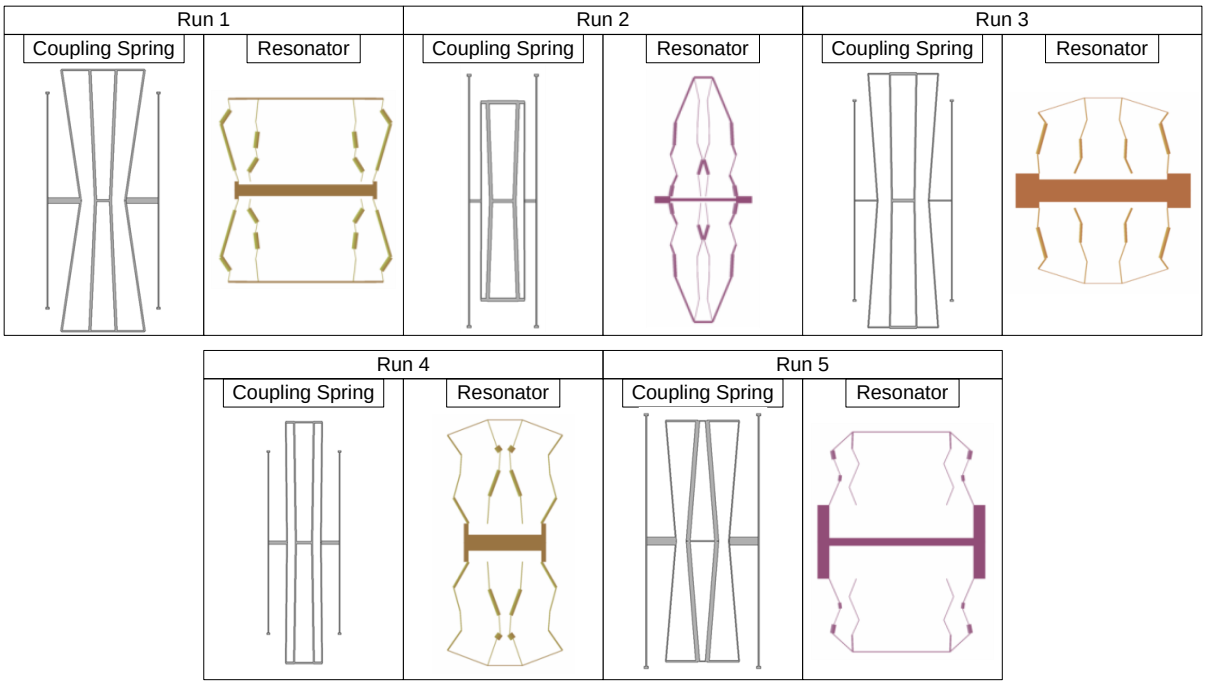


Figure 7.43 Physical - Device level run 1 – 5 coupling spring and folded flexure best result ranked by mechanical error objective for NSGAII multidisciplinary optimisation

Table 7.34 Physical – Device Multidisciplinary Optimisation Best Results Ranked by Mechanical Error

Physical – Device Multidisciplinary Optimization NSGAII				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Mechanical Error	von Mises Stress (MPa)
1	9	0	0.348674966	0.008678499
2	6	0	0.038271459	0.127508963
3	8	0	0.240896653	0.001522736
4	5	0	0.186334239	0.0003079048
5	10	0	0.0226863649	7.993169360
Physical – Device Multidisciplinary Optimization SPEA2				
Test	No of Pareto Sol in Final Pop	No Sol ≤ 1% Error Per Obj	Mechanical Error	von Mises Stress (MPa)
1	2	0	0.0261814729	0.00061076332
2	8	0	0.10298407	0.0001852307
3	9	0	0.042553795	3.512686759
4	5	0	0.0525125777	0.008400789
5	9	0	0.061800208	0.12021959378

Table 7.35 Physical - Device Level Hypervolume Results for NSGAII and SPEA2 Multidisciplinary Optimisation

Physical – Device Multidisciplinary Optimisation		
Hypervolume	NSGAII	SPEA2
S^U	397.677992567395	397.904070916963
S^M	386.5041622	395.2107632
S^L	372.098989127511	391.760671417928

* (S^U S^M S^L) [5, 80]

The application of a multidisciplinary optimisation strategy to solve the physical – device level design problem has been undertaken successfully. In both the NSGAII and SPEA2 set of results there contain highly successful results with near optimal synthesis values for both the coupling spring and folded flexure resonator. The population sets in figures 7.42 and appendix figures D.20 and D.21 show a reduction in the overall von Mises stress levels in comparison with the multi-level evaluation strategy though still very much higher than the single level strategy. The majority of the experimental runs are able to breach an mechanical error value of 0.1 equivalent to 10% total synthesis error and all runs show at least half to two thirds less mechanical error than those found at the single level strategy.

The objective values for the best results ranked by mechanical error in table 7.34 also highlight the success of the strategy to find solutions which show both good mechanical error and reduced von Mises stress levels though perhaps not to the same level as those found using the default single level strategy. Looking at the hypervolume performance in table 7.35 shows as with the previous multi-level strategy a marked increase in performance over the single level strategy. There is a slight reversal in both the NSGAII and SPEA2 algorithms on whose performance is better with SPEA2 showing overall better performance than NSGAII a change from the previous multi-level strategy. The generational hypervolume values are harder to quantify in terms of performance given that the system level population in which they are calculated from is updated only sporadically over the design process and once again the effect of the nadir point may reduce the impact of low mechanical error solutions has on the hypervolume value. Nevertheless in both the NSGAII and SPEA2 algorithms there are an increase in performance though not as noticeable in the SPEA2 result.

Looking into more depth at the mechanics of the multidisciplinary optimisation strategy as applied to the physical – device level problem, a breakdown of the mechanical error values for the best results from table 7.34 into their constituent parts are shown in table 7.36. The contribution of the coupling spring and folded flexure components and their associated synthesis error percentage values help to highlight the elements that contribute to synthesis error optimisation.

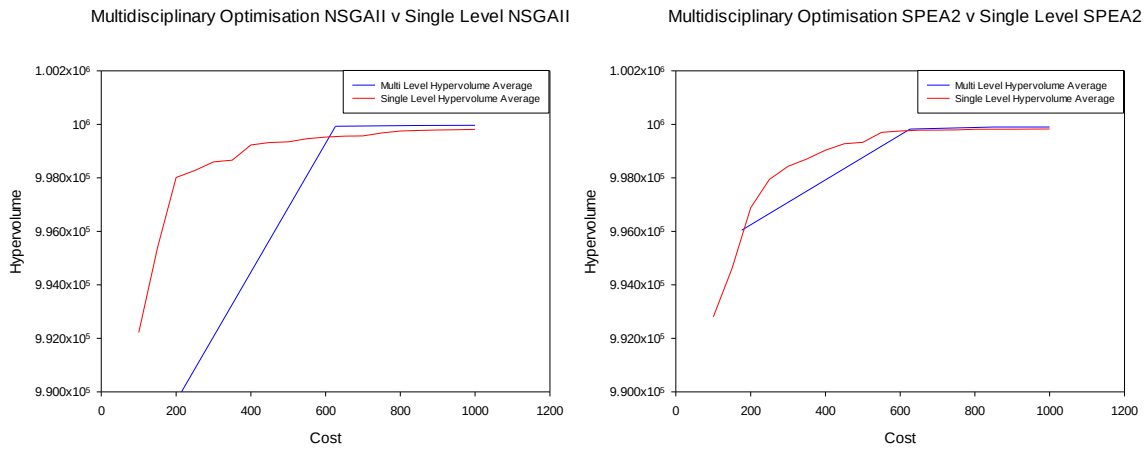


Figure 7.44 Physical - Device level average hypervolume results for the 5 runs of the multidisciplinary optimisation and single level NSGAII and SPEA2 strategies * ($S^U S^M S^L$) [5000, 200]

Table 7.36 Physical – Device Multidisciplinary Optimisation Synthesis Error % Values for Best Results by NSGAII and SPEA2 Ranked by Mechanical Error

Physical – Device MDO NSGAII				Physical – Device MDO SPEA2		
Test	Folded Flexure Stiffness Kx Error %	Folded Flexure Mass Error %	Coupling Spring Stiffness Error %	Folded Flexure Stiffness Kx Error %	Folded Flexure Mass Error %	Coupling Spring Stiffness Error %
1	1.6507	1.8415	31.375	1.5695	0.2228	0.8259
2	0.0313	1.2444	2.5515	7.1207	2.2153	0.9624
3	2.1914	0.0974	21.801	1.3062	0.1428	2.8064
4	5.1824	0.2009	13.250	1.9218	1.0313	2.2981
5	0.7803	0.1717	1.3166	3.8972	1.9432	0.3396

In particular for the solutions which contain the highest level of mechanical error, for example run 1, 2 and 3 for the NSGAII algorithm often do so as a result of the coupling spring component and the inability to reach lower synthesis error levels found more readily on the folded flexure optimisation. This can probably attributed to the lower number of functional evaluations tasked to evolving and analysing the coupling spring component where as the folded flexure resonator optimisation section of the multidisciplinary optimisation schema has a larger number of evaluations open to it to evolve better designs.

Overall the folded flexure resonator synthesis values are improved upon significantly when compared against the single level strategy and slightly when looking at the previous multi-level evaluation strategy. The optimisation of the coupling spring component mirrors this statement as well though perhaps with some solutions containing higher levels of synthesis error than is found using the multi-level evaluation strategy.

The multidisciplinary optimisation strategy looks to decompose the design problem and through an object decomposition the individual components themselves into a number of smaller and hopefully simpler optimisation tasks. In this particular schema the coupling spring is broken down into two subsystems, one and two, with a focus upon locally optimising the coupling spring stiffness kx error and von Mises stress respectively. The folded flexure resonator is also decomposed into two separate subsystems, three and four, once again each focusing upon one of the synthesis target goals for the resonator. By allowing each subsystem to focus upon a specific task, in this instance a particular objective or contributory component for an objective of the default problem it was hoped would improve the overall optimisation of the design problem.

Figure 7.45 tracks data on the synthesis error % and von Mises stress values for the best solutions found within the system and subsystem populations over the course of the design process over the 5 runs of SPEA2. The best solutions are chosen locally with the system level ranked by mechanical error, subsystem one by coupling spring stiffness kx error, subsystem two by von Mises stress, subsystem three by folded flexure mass error and finally subsystem four by the folded flexure stiffness kx error. It indicates how successful each system or subsystem branch is in optimising solutions to match local objectives while also analysing how the multidisciplinary optimisation strategy can aid in evolving solutions with an overall lower mechanical error.

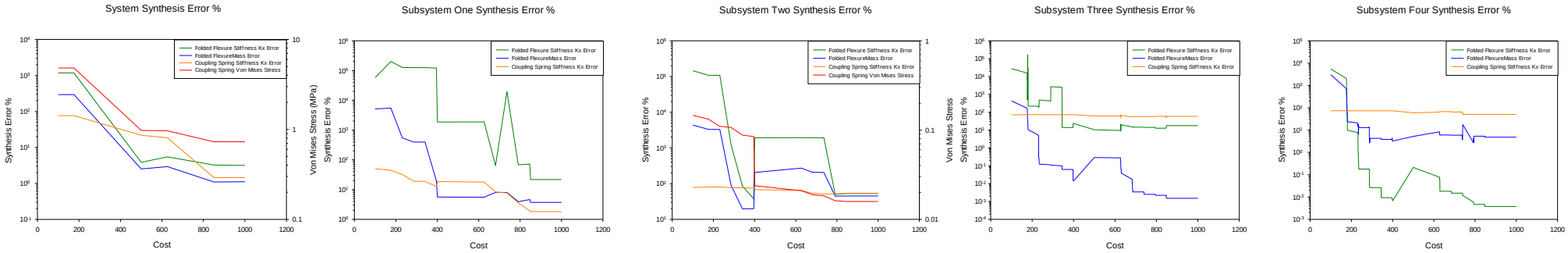


Figure 7.45 Generational synthesis error % for the best solution found over the 5 SPEA2 runs for the system and subsystem population sets. The best solutions chosen are ranked by local objectives, with mechanical error, coupling spring stiffness kx error, von Mises stress, folded flexure stiffness kx error and folded flexure mass error associated with the system and subsystem one, two, three and four respectively.

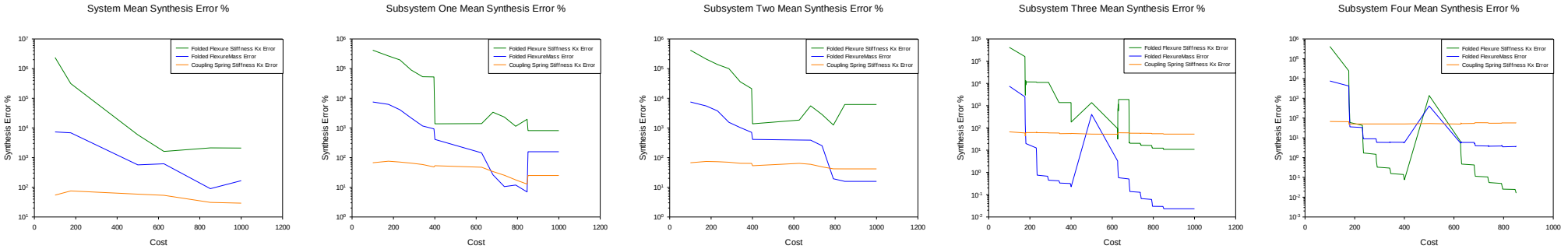


Figure 7.46 Generational synthesis error % for mean population values over the 5 SPEA2 runs for the system and subsystem population sets.

Each of the subsystems with their focus upon the individual coupling spring and folded flexure components is able to evolve solutions which have on average a local synthesis error of 1% or less with the exception of subsystem one coupling spring stiffness kx synthesis error of around 2%. Subsystem two focuses upon minimizing the von Mises stress values and is shown to continually improve such values throughout the design process though perhaps once again not to the extreme levels found by the single level strategy.

Moving on to what effect each subsystem and their evolved populations have in disseminating good building blocks associated with low synthesis error % at first glance at the subsystems associated with the coupling spring we find that solutions within subsystem one and two also show lowered synthesis error for those components under control of the other subsystems. Both subsystems one and two show a lowering of the folded flexure mass and stiffness synthesis error over the design process. The decrease in folded flexure resonator mass synthesis error seems to have been transferred reasonably well with an error on average around 3% for subsystem one, though slightly less successfully with the stiffness kx synthesis error at around 20%. Subsystem two on the other hand contains much higher values for both the folded flexure resonator stiffness kx and mass synthesis error at around 45% to 50%. Such a discrepancy can perhaps be put down to newly updated and optimised solutions from subsystems three and four no longer being able to replace established solutions with a lower von Mises stress value in subsystem two. This can in part be due to an overall trend of the design process to evolve long flexible flexure beams for the coupling spring due to subsystem one local objectives and control of key variables associated with this topology.

Focusing upon subsystems three and four and the role of optimizing the folded flexure resonator there is an improved picture with both subsystems having low synthesis error from solution building blocks transferred from other subsystems, with subsystem three having an stiffness kx synthesis error of 15% and subsystem four having a mass synthesis error of 5% at the end of the design process. The transfer of good coupling spring stiffness kx error building blocks is modest with a reduction of 73% error at the start of the process to around 55% at the end for the best solutions found locally.

The overall effect on the default system level population is a continual improvement in synthesis error values for both the folded flexure resonator and coupling spring component of the design problem to near optimal values of 1% target error. Expanding on the role of system and subsystem branches to optimise the overall synthesis error of the design problem is figure 7.46 which again looks at the synthesis error % for SPEA2 but in this case plots the mean values of each population set.

The system and subsystem population sets once again show strong improvement throughout the design process to their local objectives with the mean synthesis error for both subsystems three and four very close to the best solutions found. The subsystems associated with the coupling spring also show improvement though the mean coupling spring stiffness kx synthesis error for subsystem one is much higher at around 20%, while subsystem two higher still at 40% though this is in common with the best solutions found at this subsystem at any rate.

Subsystems one and two show much higher folded flexure synthesis error for each population on average though there is still improvement from start to finish. In contrast subsystem three

and four seem able to transfer more successfully improvements in synthesis error for the folded flexure stiffness kx and mass error respectively.

A large part in why subsystems three and four contain lower synthesis error, in particular associated with the folded flexure resonator comes naturally form the increase in functional evaluations and cycles dedicated to each subsystem over the design process. Subsystems one and two are called every ten cycles while subsystems three and four are called sequentially giving them ten times the number of functional evaluations and importantly gene transfer. The increase in functional evaluations and evolutionary process allows better solutions to be evolved, more convergence within the population to such solutions and therefore an increase in the likelihood of beneficial building blocks transferring to the other subsystems.

One disadvantage of the disparity between the coupling spring subsystems one and two and the folded flexure subsystems three and four with regards to overall functional evaluations and cycle count is the possibility of poor coupling spring designs increasing in subsystem three and four populations rather than decreasing. Because the coupling spring subsystems are called only every ten cycles if a poor design is successfully coupled with good folded flexure subsystem designs, then it is possible for the next ten cycles in the folded flexure subsystems to propagate and converge to such solutions and increase the number of poor coupling spring designs. These may then be passed on to the system level and reduce overall performance through poor coupling spring designs.

Figures 7.45 and 7.46 give a loose overview of the multidisciplinary optimisation performance and the effect of each subsystem however it is hard to draw strong conclusions to the effect that such a strategy is successful in optimising and transferring good building blocks between one or more solutions. Figure 7.47 gives a direct observation of such a phenomenon, as a solution is evolved throughout the entire design process for the best solution found in run 5 NSGAII. To begin with all four offspring solutions from each of the subsystems are brought together at cycle 11 through the gene swapper module where their individual design variables are transferred / collated into a single solution which remains in subsystem three. It is a clear example of both local transfer of the decomposed coupling spring and folded flexure components in this case improving the coupling spring stiffness error significantly, and of the whole components themselves, bringing together a good coupling spring with a good folded flexure resonator.

This solution is evolved through further crossover, mutation and gene transfer over a number of cycles between subsystems three and four bringing down the folded flexure resonator synthesis error with an example of gene transfer shown in cycle 32 producing the solution present at cycle 51 of the subsystem three population. This solution itself is then successfully transferred to the system population where once again local variation is able to improve it further. The complete process is able to bring solutions which contained a synthesis error ranging from over 100K% to 200% and optimise them into a final solution containing only 2% synthesis error.

Figure 7.47 Genotype / Phenotype generational spanning trees for best synthesis error % solution for NSGAII MDO run 5.

7.2.6 Physical – Device level comparison and analysis

Microelectromechanical systems can consist of a number of individual components or devices which together communicate and function in unison to provide an established behaviour set out by the designer. Often such devices or components hail from a number of different disciplines and fields which can require separate and specific disciplinary modelling and analysis.

The hierarchical modelling and analysis present in MEMS provides both challenges and opportunities in their design optimisation in particular using automated soft computing methods. It is wholly conceivable that designers face modelling and analysing such MEMS devices which consist of two or more components that span more than one hierarchical level as constructed within the physical – device level design problem. This particular problem was used to investigate the effect of optimising a MEMS device which contained highly computationally expensive solid modelling and FEA at the physical level of MEMS design coupled with a lower device level NODAL modelling analysis component.

It may not always be possible to synthesis and optimise each component of a MEMS device separately as analysis and constraints tied to one particular component modelled at one level may be linked or required for modelling and analysis of another component at another level of design. Here the two components a physical level coupling spring model and a device level folded flexure resonator are connected through there linked function within a MEMS bandpass filter device. The force and displacement placed upon the coupling spring device is linked with the modelling and analysis of the folded flexure resonator model and the comb drive transducer within it. The task of optimising the von Mises stress placed upon it is therefore linked both to the structure of the component itself but also the force analysis retrieved from the folded flexure model as well.

In situations such as these there come challenges that arise from coupling together the two modelling and analysis levels, in particular the benefit of fast analysis using NODAL modelling tools is lost due to a fixation upon the constraints of computational cost from physical level FEA. This can reduce the amount of functional evaluations that can be spent in synthesizing and optimising the device level components than would possibly have occurred under a separate optimisation strategy.

Table 7.37 Physical – Device Level Hypervolume Results for Single and Multi-level Strategies for Both NSGAII and SPEA2 Algorithms.

Physical – Device Level NSGAII			
Hypervolume	Single Level	Multi-Level Evaluation	Multidisciplinary Optimisation
S^U	338.087859942512	396.452937469177	397.677992567395
S^M	325.1325906	387.3681500	386.5041622
S^L	304.048705831891	380.342463184353	372.098989127511
Physical – Device Level SPEA2			
Hypervolume	Single Level	Multi-Level Evaluation	Multidisciplinary Optimisation
S^U	349.351035125636	393.597901525275	397.904070916963
S^M	331.3561860	385.1080075	395.2107632
S^L	320.808824069444	376.836837582899	391.760671417928

* (S^U S^M S^L) [5, 80]

The hypervolume results for the single and multi-level strategies are shown in table 7.37 with a clear distinction in performance between the single level strategy and the multi-level strategies employed to solve the physical – device level design problem. This isn't surprising given the inability of the single level strategy to optimise both the coupling spring and folded flexure components together to have a low mechanical error.

The two multi-level strategies across both the NSGAII and SPEA2 algorithms seem split with multi-level evaluation performing better for the NSGAII algorithm while the multidisciplinary optimisation strategy performs superior within the SPEA2 results. A closer look at the values probably show the MDO strategy to be the better of the two with close mean values for the NSGAII algorithm given the effect of having a much lower bound value, while the multi-level strategy for SPEA2 is much farther away in both mean and boundary values to that of the MDO strategy.

Still the values themselves can be biased with the chosen nadir point for calculation, so it is good practice to look over all the results for both strategies. Looking back the final best results for both strategies show MDO to more consistently provide solutions with a mechanical error lower than 10% and more often contain lower levels of stress. Both strategies seem able to optimise the coupling spring and folded flexure components to have low synthesis error with perhaps a slight boost in performance towards optimising the folded flexure resonator by the MDO strategy.

7.3 Summary

The application of the current state of the art in automated MEMS design synthesis and optimisation along with a set of multi-level strategies have been successfully applied to a number of bi level design problems spread throughout the design hierarchy of MEMS. The MEMS design problems focused upon two levels of modelling and analysis that were coupled together in some form. A system – device problem design a device level bandpass filter while analysing function at the system level, and a physical – device problem involving two separately analysed components linked in their function together.

The coupling of both system and device modelling levels allowed for a different approach to the design optimisation of a MEMS bandpass filter which showed more robustness and higher performance than the 'de-coupled' uni level approach. The multi-level strategies proved successful once again, though with lower performance for the multi-level parameterisation schema.

The final design problem focused upon the physical layout of the bandpass filter and the components that make up the whole system. The modelling and analysis at the physical level brings with it many challenges for automated design, in particular the high computational cost. The ability to reduce the cost of analysis through the application of a multi-level strategy brought a significant improvement to the overall designs at the end of the optimisation process, while applying a multidisciplinary optimisation strategy to decompose the problem into a number of simpler design tasks provided superior performance to all other strategies.

The next chapter brings together all three levels of modelling and analysis into a single MEMS design problem and looks to evaluate each single and multi-level strategies outlined.

8

Tri-Level Design Optimisation

This chapter looks to accumulate the two previous chapters work into a three level design problem which looks to couple together the system, device and physical levels of modelling and analysis into a coupled tri level MEMS design problem. This involved the synthesis and optimisation of a bandpass filter consisting of both device level and physical level components and analysed using a system level electrical equivalent model. Both the single level, multi-level and multidisciplinary optimisation strategies are applied and compared.

The design synthesis of MEMS in particular the aspects that cover both modelling and analysis has been outlined into a number of levels in which a designer can choose to engage and provide suitable mediums to build and optimise such a device. One particular hierarchical structure was outlined by Senturia [16] and breaks up the design process into four distinct modelling and analysis levels (System, Device, Physical and Process). The first three levels capture the modelling and analysis associated with the direct function and behaviour of the device while the final process level itself encompasses process for fabrication for example mask layout design.

As a designer opening up all modelling and analysis levels to the design synthesis of a MEMS device may both be beneficial but also a necessity with the requirement of more than one level needed to provide full analysis of a device. The previous chapter provided both an example of a coupled design problem between the system and device levels where such a construction allowed system level analysis to be coupled with device level modelling, and a physical and device modelling and analysis between two strongly coupled components of a much larger device.

The final fully coupled design optimisation problem presented here looks to include all three levels of design into a single task taking a similar approach to the system – device problem by optimising the components and layout of a MEMS bandpass filter. In addition to this the coupling spring capacitance variable is replaced with a fully functioning coupling spring solid model and a third objective linked to the component is added. Overall as seen in figure 8.1 a system – device – physical level problem for bandpass filter design synthesis has been constructed which looks at the role each level of design plays and contributes to the design synthesis of a MEMS device.

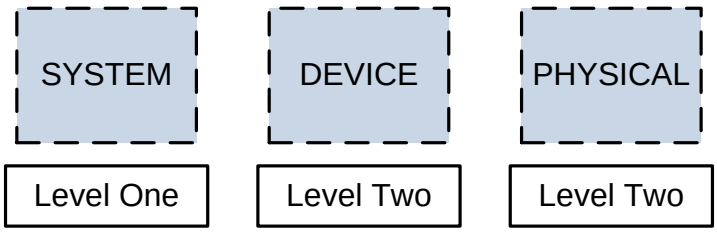


Figure 8.1 Tri level design optimisation of MEMS using multi-level design strategies

8.1 System – Device – Physical level design optimisation

The system – device – physical level design problem is the accumulation of all previous modelling and analysis undertaken to synthesis a MEMS bandpass filter through the coupling of the device level folded flexure resonator and physical level coupling spring components and then analysing there filter response through system level circuit models.

The mechanical elements of the bandpass filter are modelled using the previous NODAL folded flexure resonator and FEA butterfly coupling spring components with the electrical equivalent circuit model built from the mechanical values derived from each of the components that make up the bandpass filter in a similar vein to the previous system – device level design problem. The design variables are listed in table 8.1 and consist of NODAL folded flexure resonator, FEA coupling spring and comb transducer model variables.

Table 8.1 System – Device – Physical Level Filter Variable Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Central Mass Length (μm)	Real Valued	10	600
Central Mass Width (μm)	Real Valued	10	400
Shuttle Yolk Length (μm)	Real Valued	2	400
Shuttle Yolk Width (μm)	Real Valued	10	400
Beam Number	Integer	1	6
Beam Angle	Real Valued	45	135
Beam Length (μm)	Real Valued	5	100
Beam Width (μm)	Real Valued	2	20
Truss Width (μm)	Real Valued	2	20
Anchor Placement Length (μm)	Real Valued	0	400
Thickness (μm)	Real Valued	2	200
Truss Beam One Length (μm)	Real Valued	5	100
Truss Beam One Width (μm)	Real Valued	1	10
Truss Beam One Length (μm)	Real Valued	5	200
Truss Beam One Width (μm)	Real Valued	1	10
Mass Connector Length (μm)	Real Valued	2	100
Mass Connector Width (μm)	Real Valued	1	20
Centre Beam Length (μm)	Real Valued	20	100
Centre Beam Width (μm)	Real Valued	2	10
Outer Flexure Beam Width (μm)	Real Valued	1	10
Inner Flexure Beam Width (μm)	Real Valued	1	10
Flexure Length (μm)	Real Valued	50	400
Centre Gap Length (μm)	Real Valued	10	50
Coupling Spring Thickness (μm)	Real Valued	1	5
Voltage	Real Valued	1	200
Tank Number	Integer	1	9
Finger Number	Integer	1	200
Comb Thickness ($\boldsymbol{\mu m}$)	Real Valued	2	30

The objectives associated with the bandpass filter remain the same with the goal to synthesize a filter transmission which matches design targets and shape set out in chapter 6. Additionally a third cost objective is added to try and minimize the overall von Mises stress values of the coupling spring components of the mechanical filter. The constraints remain the same for each of the components used in previous experiments and are also shown in table 8.2.

Table 8.2 System – Device – Physical Level Filter Problem Information

Objectives		Constraints	
Bandpass Filter Response Error	Minimize	Tank Constraint 1 [Truss Beam Two Length – (Inner Flexure Beam Width * 2)]	Inequality ≥ 0
Bandpass Central Frequency Error	Minimize	Tank Constraint 2 [Centre Beam Length – ((Inner Flexure Beam Width * 2) + 3e-06)]	Inequality ≥ 0
von Mises Stress	Minimize	Tank Constraint 3 [((Centre Beam Length / 2) + Centre Gap Length + Mass Connector Length) – ((Truss Beam Two Length / 2) + Truss Beam One Length + 10e-06)]	Inequality ≥ 0
		Tank Constraint 4 [Shuttle Yolk Length - Central Mass Width]	Inequality ≥ 0
		Intersection Check	Inequality

The algorithmic parameters for the population sets used in both NSGAII and SPEA2 are shown in table 8.3 and are restricted in size due to the high computational cost from the physical level analysis. The addition of the NODAL folded flexure resonator model means the inclusion of the structural crossover module tied to the folded flexure spring component of the resonator into the design pathway with the remaining marked variables undergoing SBX crossover. The force applied to the coupling spring component for von Mises stress analysis is once again derived from the comb transducer component of the updated resonator model.

Table 8.3 System – Device – Physical Level Algorithmic Parameter Values

Algorithm Parameter	Default Value
Population Size	50
Offspring Size	50
Selection Size	50
Replacement Size	100
Generations	20

The number of functional evaluations over the whole design process is also restricted to 1000 due to the computational cost of the physical level coupling spring analysis which takes up the majority of analysis cost for each solution. The single level strategy for solving the system – device – physical level design problem is shown in figure 8.2 along with the representation used and node markers and structural tags for tank and beam units.

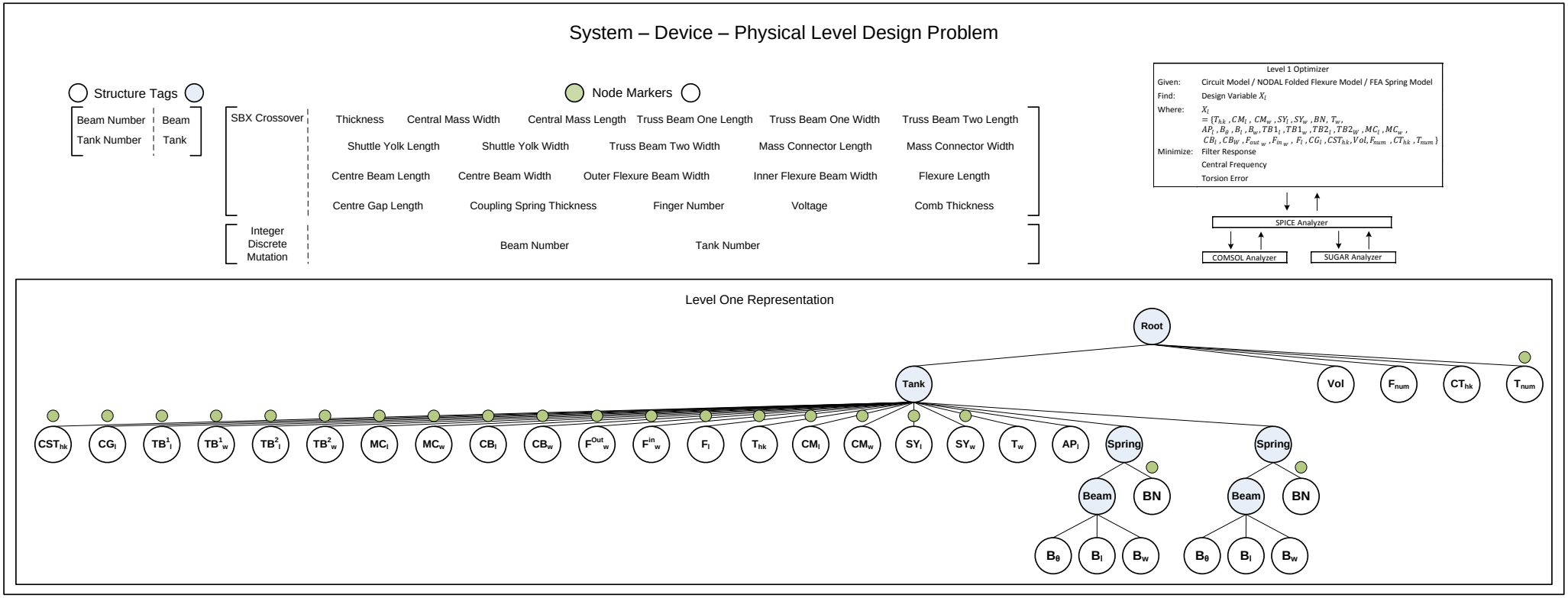


Figure 8.2 System – Device – Physical level design template, with overview of problem, default representations, structure tags and node markers.

8.1.1 Multi-level evaluation

The multi-level evaluation strategy applied to the system – device – physical design problem is an amalgam of previous evaluation strategies used throughout the design problems of chapters 6 and 7. As in the system – device level multi-level evaluation construction two separate levels of modelling accuracy and cost are used as seen in figure 8.3 with the lowest level of accuracy and cost consisting of reduced filter frequency circuit analysis and the use of analytical and nodal modelling for the folded flexure resonator and coupling spring respectively. The highest or default level of analysis and cost remains unchanged from the single level construction with full circuit analysis, and the use of the more complex NODAL folded flexure resonator and FEA coupling spring models.

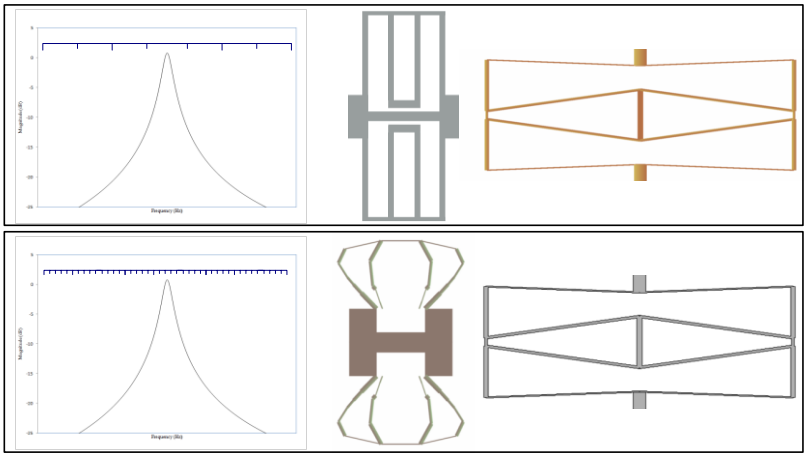


Figure 8.3 Multi-level evaluation level one and two models and analysis calls with level one (top) reduced sampling circuit analysis, analytical folded flexure resonator and NODAL coupling spring models and level two (bottom) full circuit analysis, NODAL folded flexure resonator and FEA coupling spring models.

Table 8.4 Evaluation costs for level one and two analysis

Full Analysis		Reduced Analysis	
Frequency Range	Sampling Size	Frequency Range	Sampling Size
1Hz – 15kHz	15,000	1Hz – 15kHz	5,000
Mean Analysis Time (20 Calls)		Mean Analysis Time (20 Calls)	
30.769344		0.693571053	
Ratio			
1 : 44.36365			

The computational cost of a single evaluation for both levels one and two are shown in table 8.4 and correspond to a solution containing five resonator tanks and therefore five separate evaluations for both the folded flexure resonator and coupling spring component. The main contribution for computational cost at the highest level comes from the FEA coupling spring giving 30 seconds of wall clock time to analysis and as a result there is a large discrepancy in overall analysis time between each level. This ratio between the two levels is floored to give 40 level one calls for every 1 level two analysis call with the direct result of the level one pathway being called sequentially throughout the design process and level two being called every 12

cycles. The total number of functional evaluation cost is still 1000 with the setup giving around 6400 individual calls at level one analysis and 850 calls for the full level two analysis. The objectives remain the same as the default single level design problem with the exception that level one no longer undergoes von Mises stress analysis and therefore the objective is not used. The constraints reflect the models used for each level with level one constraints related to the analytical folded flexure resonator and NODAL coupling spring models and level two remaining unchanged from the default problem. Table 8.5 holds level one objective and constraint information for the multi-level evaluation strategy.

Table 8.5 System – Device – Physical Multi-Level Evaluation Level One Problem Information

Objectives		Constraints	
Bandpass Filter Response Error	Minimize	Tank Constraint 1 [Truss Beam Two Length – (Inner Flexure Beam Width * 2)]	Inequality ≥ 0
Bandpass Central Frequency Error	Minimize	Tank Constraint 2 [Centre Beam Length – ((Inner Flexure Beam Width * 2) + 3e-06)]	Inequality ≥ 0
		Tank Constraint 3 [((Centre Beam Length / 2) + Centre Gap Length + Mass Connector Length) – ((Truss Beam Two Length / 2) + Truss Beam One Length + 10e-06)]	Inequality ≥ 0
		Tank Constraint 4 [Shuttle Yolk Length - Central Mass Width]	Inequality ≥ 0
		Tank Constraint 5 (([Beam Width] * 4) - ([Truss Length] * 3)) + 9e-6	Inequality ≤ 0
		Intersection Check	Inequality

The introduction of the separate levels requires the use of migrator modules to transfer successful solutions between each level periodically. Table 8.6 holds the migration module parameters for both levels with a significant change of selecting 100% of the local population set for transfer given the low number of solutions present to begin with. This should hopefully boost the number of solutions which successfully transfer from one level to another though with a risk of premature convergence.

Table 8.6 Migrator Module Parameters for Multi-Level Evaluation

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	100	12
Level 2	Level 1	100	12

Transferring solutions from one level to another also brings with it the requirement to convert the analytical folded flexure resonator model representation to that of the nodal model representation. Therefore the migration module is tied with a subtree converter module to aid in this conversion process which follows the same schema set out for the device multi-level evaluation process to allow natural and easy transfer. Figure 8.4 outlines the overall multi-level evaluation strategy for solving the system – device – physical design problem, and includes both low and high level representations and their associated variables, structure tags and node markers.

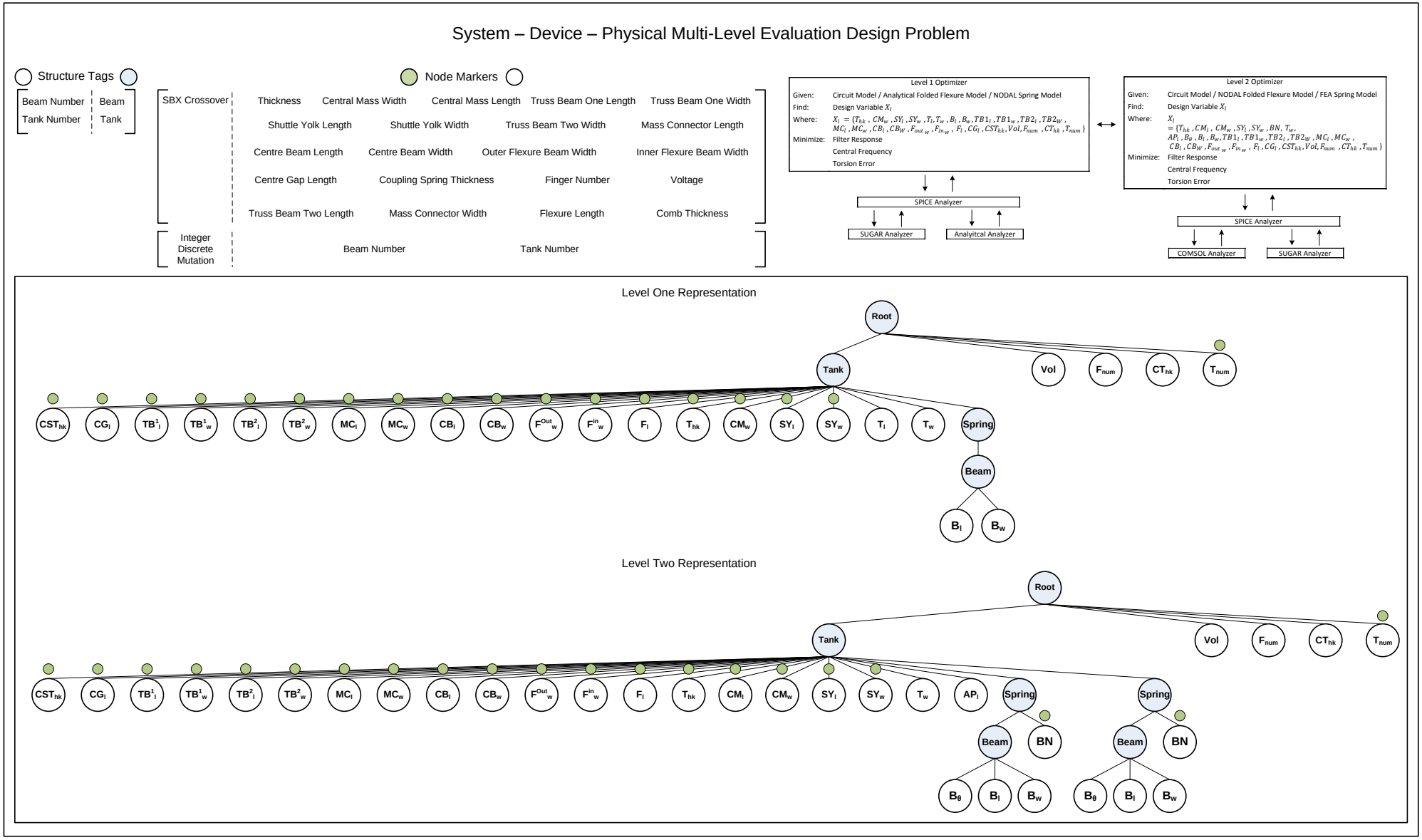


Figure 8.4 System – Device – Physical multi-level evaluation design template, with overview of problem, default representations, structure tags and node markers.

8.1.2 Multi-level parameterization

One of the stark changes introduced with the system – device – physical level design problem is the growth in decision variables present in the solution representations. The original system level instantiation contained only 19 design variables for the electrical equivalent circuit model representation containing five tanks. Here the same number of tanks gives rise to 159 variables for an average of three beams for each resonator spring of the NODAL folded flexure model. Clearly aiding the optimizer in handling such a large number of decision variables can be beneficial, in particular the multi-level parameterization strategy tries to do so by managing the number of variables open to the optimizer over a number of levels.

Table 8.7 System – Device – Physical Multi-Level Parameterization Level Variable Count

	Level One		Level Two		
	1 Tank / 1 Beam	3 Tanks / 3 Beams	1 Tank / 1 Beam	5 tanks / 3 Beams	9 tanks / 6 Beams
Variable Count	25	82	32	204	526

The multi-level parameterization strategy employed at the system – device – physical level follows the similar approach in the system and system – device parameterizations through cloning of individual decision variables over each tank of the bandpass filter. This is provided by having two levels of parameterization with level one cloning the variables for the coupling spring and folded flexure resonator models. In addition the number of variables open to variation for the coupling spring model has been reduced from 20 to 6 in what is similar to the previous physical level parameterization method in chapter 6.

The folded flexure spring component has also had its variation scope reduced by fixing the beam angle to 90 degrees and having the beam length and number as global cloned variables so only one beam number and beam length variable is active for the representation.

The final change at this level is the reduction in the number of tanks that can be added to the bandpass filter by reducing the upper bound from its default value of 9 to 3. The highest level of parameterization, level two is simply the unchanged default representation from the single level approach. The effect of these changes on the number of design variables open to the optimizer is shown in table 8.7 with a moderate to large reduction between the two levels.

Table 8.8 Migrator Module Parameters for Multi-Level Evaluation

Migration Level	Destination Level	Migration Percentage	Cycle Count
Level 1	Level 2	20	2
Level 2	Level 1	20	2

Table 8.9 System – Device – Physical Multi-Level Parameterization Level One Filter Problem Information

Variable Tag	Sub Tree Type	Lower Bound	Upper Bound
Tank Number	Integer	1	3
Beam Angle	Real-Valued	90	90

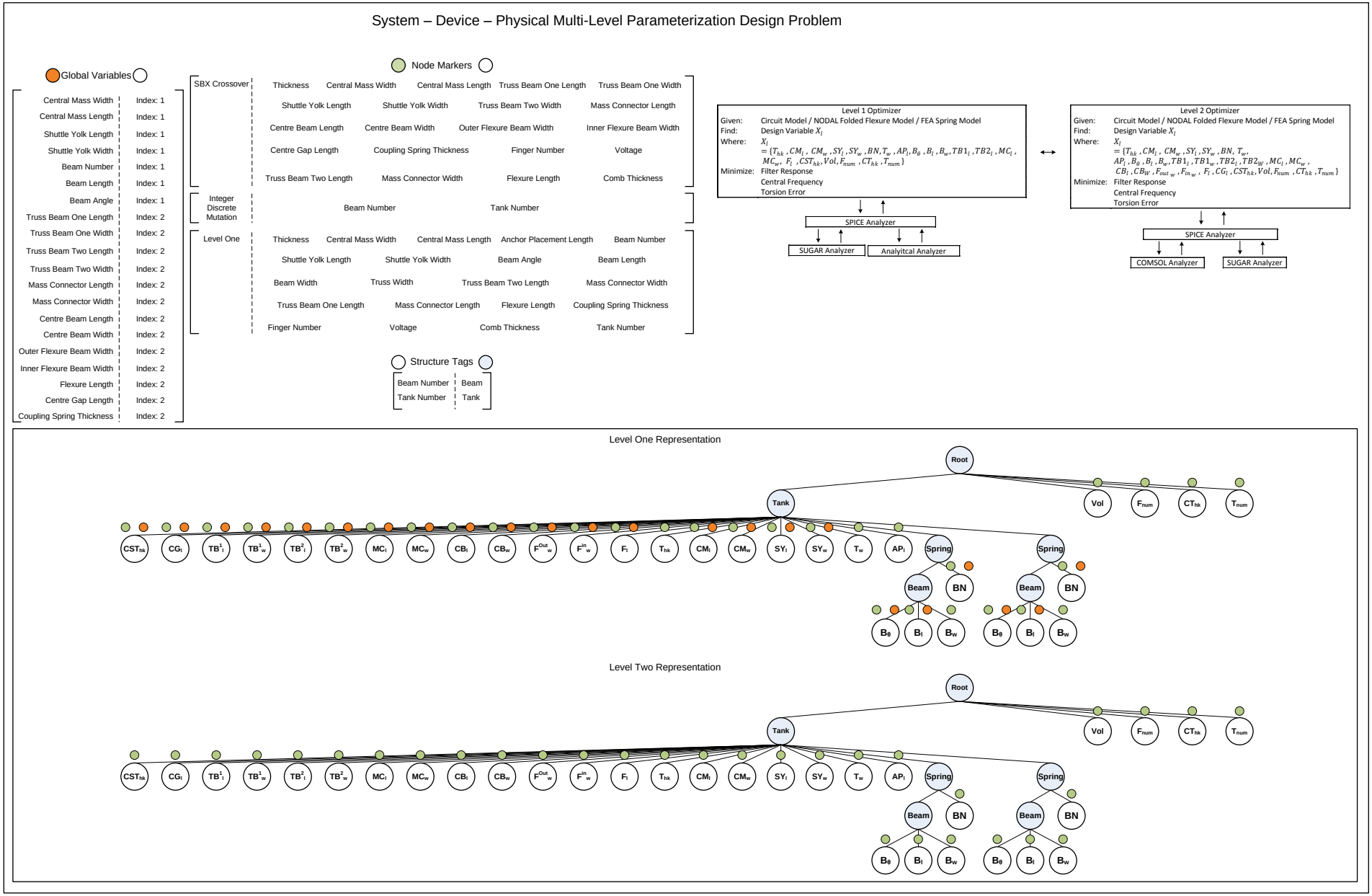


Figure 8.5 System – Device – Physical multi-level parameterization design template, with overview of problem, default representations, structure tags, global and node markers.

The addition of the two levels of parameterization requires the use of migration modules to transfer solutions and where necessary convert the various components and their variables to their level specific values. Transfer from level two to level one requires a number of variables to be marked as global while also altering the bounds on the folded flexure spring beam angle variables to match the value set out for this level and setting the tank number upper bound to three, solutions which have more than 3 tanks have their number clipped to match the new upper bound values. Solutions moving from level one to two simply require removing the global markers placed upon the individual variable and removing the upper bound restriction on the beam angle and tank number variables to match the default parameter values. The migrator module parameters are shown in table 8.8, while the level one parameter bound changes are shown in table 8.9.

The multi-level parameterization strategy is outlined in figure 8.5 and includes the representations for the simplified and default parameterizations and their associated global variables, structure tags and node markers.

8.1.3 Multidisciplinary optimisation

The final strategy to be applied is multidisciplinary optimisation, based upon the same approach found in chapter 6 and the system level design problem with a decomposition of the filter transmission into a number of objectives tasked to improve certain characteristics of the bandpass filter transmission shape as seen in table 8.10.

The default system level design problem is decomposed into two separate subsystems with each one focusing on specific objectives and under control of specific decision variables tied to these objectives through their influence. The first subsystem contains the decision variables associated with the folded flexure resonator component which are linked heavily with the characteristics of the pass band error objective and the filter frequency of the bandpass as a whole [131][274]. The second subsystem focuses upon the topology of the bandpass filter and the coupling spring component with the task of maximizing the bandwidth while reducing the stop band error all the while trying to minimize the total von Mises error of the filter. The coupling spring variables naturally control the level of von Mises stress within the filter while also influenced by the number of tanks present within the filter as well. This goes against the aim of trying to reduce the stop band error of the filter transmission shape through increasing the number of tanks and subsequent increase in drop off of the filter shape [131] and therefore may affect performance. The system level contains the default objectives for the design problem and focuses upon the design variables which play an influence upon both subsystems these being the voltage, finger number and thickness of the comb transducer component of the filter.

Table 8.10 System – Device – Physical Multidisciplinary Optimisation Objectives

	System Level			Subsystem 1		Subsystem 2		
	Objective 1	Objective 2	Objective 3	Objective 1	Objective 2	Objective 1	Objective 1	Objective 3
Objective Type	Minimize	Minimize	Minimize	Minimize	Minimize	Minimize	Maximize	Minimize
Objective Description	Filter Response	Central Frequency	Von Mises Stress	Pass Band Error	Central Frequency	Stop Band Error	Bandwidth	Von Mises Stress

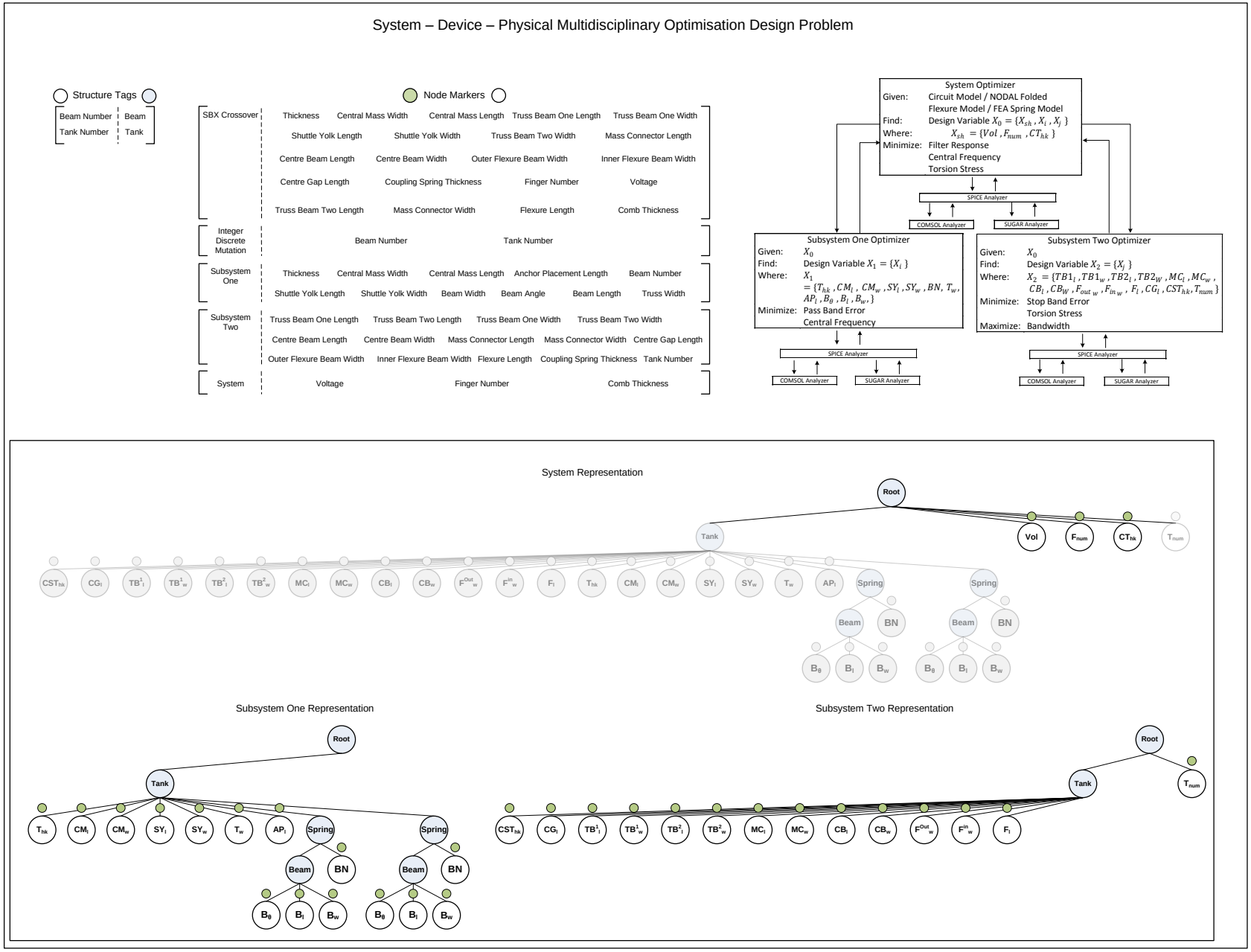


Figure 8.6 System – Device – Physical multidisciplinary optimisation design template, with overview of problem, default representations, structure tags and node markers.

Table 8.11 System – Device – Physical Level Multidisciplinary Algorithmic Parameter Changes

Algorithm Parameter	Default Value
System Population Size	25
System Offspring Size	25
System Selection Size	50
System Replacement Size	50
Grand Pareto Size	75
Subsystem Population Size	25
Subsystem Offspring Size	25
Subsystem Selection Size	25
Subsystem Replacement Size	50
Subsystem Total Size	50

The multidisciplinary optimisation strategy decomposes the problem into a number of subsystems each of which requires a separate population set. Though the entire design process still utilizes 1000 functional evaluations in order to split this efficiently between both system and subsystem populations the size of each has been reduced. Table 8.11 contains the parameter values for both the system and subsystem population sets with a reduction in the default population from 50 to 25 so as to allow more search to be undertaken at the subsystem levels. The multidisciplinary optimisation strategy is outlined in figure 8.6 and includes the representations and their associated global variables, structure tags and node markers.

The design process involves running the subsystem levels sequentially while calling the system level every 4 cycles and as a result this gives around 250 functional evaluations at the system level and 750 functional evaluations at the subsystem level.

8.2 System – Device – Physical level numerical results

The system – device – physical level design problem is the accumulation of the previous design optimisation of various MEMS components and systems into a final complex hierarchical design problem. Each of the single and multi-level design optimisation strategies has been applied to solve this particular design problem, the synthesis and optimisation of a bandpass filter. The objective values for the best solution found ranked by the filter response error for each strategy is shown below in table 8.12 for both NSGAII and SPEA2 algorithms.

The hypervolume results for each strategy are shown in table 8.13 with the best and second best results for the mean and bound hypervolume values shaded for both NSGAII and SPEA2 algorithms. Finally the generational hypervolume results for both NSGAII and SPEA2 of the single level strategy and SPEA2 for each of the multi-level strategies are shown in figure 8.7.

The design optimisation of an electrical equivalent bandpass filter circuit model at the system level of hierarchical MEMS design was the beginning of an exploration into the role of how multi-level design optimisation strategies can be employed to help automate the process of MEMS design. From this simple example aspects of the components that make up the bandpass filter have been expanded with modelling and analysis from higher more complex levels of design also incorporated to eventually give an example of a design problem which spans all three levels of MEMS design.

Table 8.12 System – Device – Physical Single and Multi-Level Bandpass Filter Results

System – Device – Physical Level NSGAII					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	5	9071.395	4520	4.11555	1
2	1	7384.547	483	44.6559	2
3	3	8348.247	20	80.7708	2
4	3	9071.445	4530	0.03577	1
5	0	7331.662	1331	7.90296	2
System – Device – Physical Level SPEA2					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	16	9078.716	4590	2.357E-05	1
2	45	5835.054	44	10.4792	3
3	47	9188.146	4220	6.61295	1
4	22	6811.317	343	9.53182	3
5	21	5628.791	2220	3.09043	2
System – Device – Physical Multi-Level Evaluation NSGAII					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	0	4640.063	211	1.37479	2
2	29	3097.612	79	1.87078	2
3	3	3012.197	30	17.3022	2
4	15	2116.169	152	5.30924	7
5	37	4072.873	102	6.80670	2
System – Device – Physical Multi-Level Evaluation SPEA2					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	18	3088.871	55	17.0536	2
2	42	2267.081	790	21.1212	2
3	24	3194.171	88	2.98859	2
4	17	4032.604	486	14.1470	4
5	13	3248.557	80	2.00902	2
System – Device – Physical Multidisciplinary Optimization NSGAII					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	4	9132.119	4300	0.00812	1
2	1	9145.008	4280	0.00291	1
3	3	9105.263	4360	3.79799	1
4	0	9481.360	4590	0.00093	1
5	5	9071.582	4520	0.63645	1
System – Device – Physical Multidisciplinary Optimization SPEA2					
Test	Index	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	12	9351.066	5000	0.12795	1
2	11	9080.331	4610	0.00359	1
3	5	5061.034	940	1.63016	2
4	8	9122.225	4340	0.02277	1
5	13	9085.407	4620	0.00071	1
System – Device – Physical Multi-Level Parameterization NSGAII					
Test	Level	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	Low	6626.988	970	8.62222	2
2	Low	7432.164	4700	27.1376	2
3	Hi	4449.858	136	12.6503	2
4	Hi	6946.686	4290	23.5390	2
5	Low	6228.612	4759	200.100	2
System – Device – Physical Multi-Level Parameterization SPEA2					
Test	Level	Filter Objective	Central Frequency Objective	Voltage	Tank Number
1	Low	7467.120	2806	7.06658	2
2	Low	5267.627	1057	2.66968	2
3	Low	3375.395	1050	1.21913	2
4	Hi	7122.999	10	8.15084	2
5	Hi	5280.925	1396	3.46564	2

Table 8.13 System – Device – Physical Level Hypervolume Results for Single and Multi-level Strategies for Both NSGAII and SPEA2 Algorithms.

System – Device – Physical Level NSGAII				
Hypervolume	Single Level	Multi-Level Evaluation	Multi-Level Parameterization	Multidisciplinary Optimisation
S^U	136290189257.835	146659287267.636	142771704555.765	122916880648.342
S^M	132291228211.432	143815480596.395	137899741544.686	122215879787.052
S^L	122652591185.954	140183929862.534	134843492071.537	121498255000.815
System – Device – Physical Level SPEA2				
Hypervolume	Single Level	Multi-Level Evaluation	Multi-Level Parameterization	Multidisciplinary Optimisation
S^U	140337132665.166	145123007927.737	143452910586.008	134075979686.322
S^M	133537741527.420	144251490174.489	140218104713.789	125118815394.729
S^L	122543356083.238	142352698156.314	137118953109.163	118697705097.625

* (S^U S^M S^L) [100000, 5000, 300]

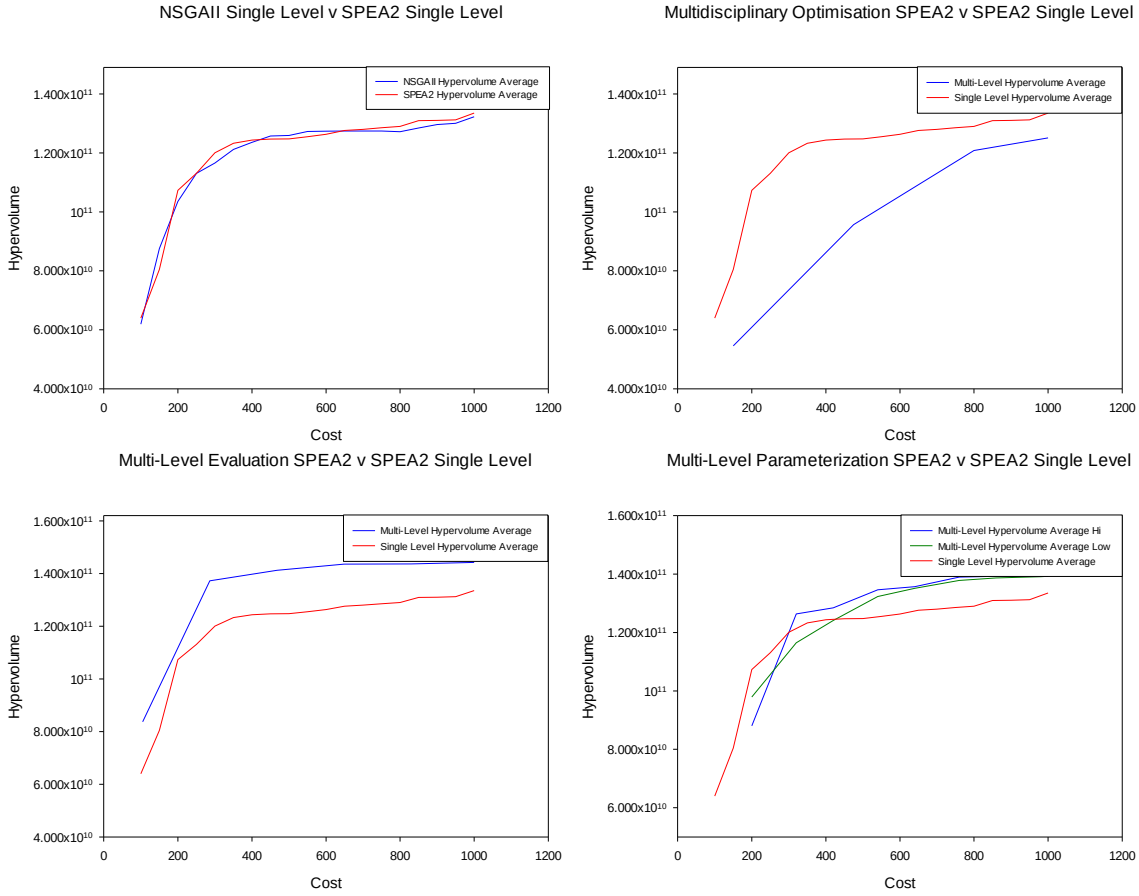
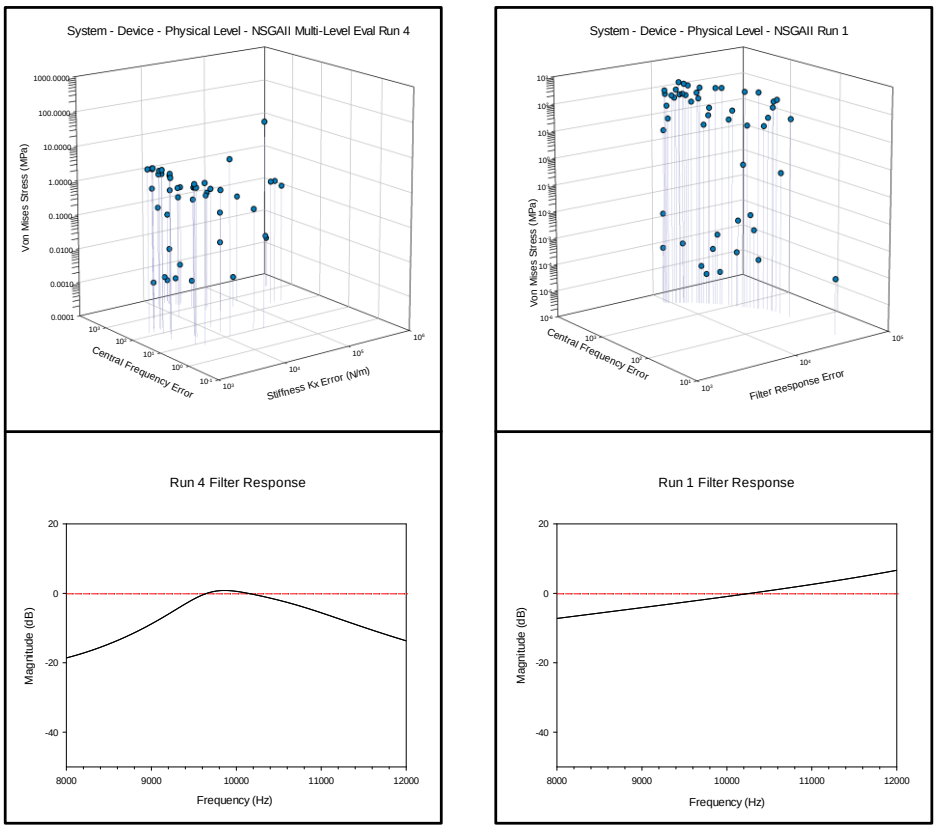


Figure 8.7 System – Device – Physical level average hypervolume results for the 5 runs of the (top right) single level NSGAII and SPEA2 strategies, and remaining SPEA2 multi-level strategies * (S^U S^M S^L) [100000, 5000, 300]

There are many differences between both approaches however they do allow some level of comparison as to the success of modelling, analysing and optimising a bandpass filter using all three levels of MEMS design open to a designer. The best filter responses found show a disappointingly high level of filter error in particular for the single level and multidisciplinary optimisation strategies with values of 8000+ common.

One of the major factors in reduced performance is simply down to the number of functional evaluations tasked to solve this particular design problem when compared against the system level design problem from chapter 6. Here only 1000 functional evaluations are used throughout the entire design process a tenth of that used previously and a direct result of the extra computational cost from analysing the physical level coupling spring model. In addition there is a new objective to minimise the von Mises stress of the bandpass filter which can lead to poor filter response error solutions remaining in the population simply due to this objective and also increased 'energy' from the optimizer also having to focus upon optimizing this objective.

The hypervolume results in table 8.13 along with the previous best filter error response solutions indicate that the best methods to solve this problem are the multi-level evaluation and parameterization strategies. Both are able to consistently find solutions which are superior in filter response to the standard results found using the single and multidisciplinary optimisation strategies with multi-level evaluation shifting even further in improvement of the bandpass filter transmission shape.



(a) (b)

Figure 8.8 System – Device – Physical level final population sets and best filter response ranked by filter frequency objective for (a) NSGAII multi-level evaluation strategy run 4 and (b) NSGAII single level run 1.

The generational plots for hypervolume corroborate with this assessment with both the multi-level evaluation and parameterization strategies outperforming the single level approach significantly. The ability for the optimiser to use a far larger number of functional evaluations

over the design process has clear advantages even given the possible side effect of the lower cost modelled solutions not transferring as successfully to the default modelling level due to differences in analysis between them. There is also the possible benefit of the lower cost level of the multi-level evaluation strategy only having to optimize two objectives therefore being able to focus entirely upon the bandpass filter transmission shape.

Figure 8.8 provides two examples of bandpass filter design found from the single and multi-level evaluation strategies with the final population sets and best filter transmission shape for both presented. The result found by the multi-level evaluation strategy represents the best solution found out of all the experiments with a reasonable bandpass shape but very little in terms of filter drop off and therefore giving a high stop band error. The filter shown in figure 8.8b represents a solution with a filter response error of around 9000+ with no discernable bandpass features which may come in some way to show the poor ability for the optimizers to evolve bandpass shapes away from this particular shape.

The combination of all three levels together to form a unique bandpass filter design problem throws up some interesting hurdles when looking to design a MEMS bandpass filter overall. The first involves whether as a designer it is better to approach designing the problem in as decomposed as a state as possible with the various components unconnected and optimised separately as demonstrated with the uni level examples in chapter 6. The formulation for the system – device – physical level design problem was under the guise that each component is inseparable with the device level modelling and analysis tied to both the system and physical level modelling and analysis. Clearly undertaking the design optimisation of a MEMS bandpass filter in such a way has not matched the level of synthesis and optimisation provided by the system and system – device level examples.

The major difference comes from the addition of the FEA coupling spring model into the representation of the design problem and the additional cost objective it brings with it. There is suddenly an additional 13 decision variables the optimizer now has to vary and for each tank that is added to the bandpass filter representation it only scales higher when compared with the system level representation which only contained a single coupling spring capacitance decision variable in its stead. This can in some way also explain how the multi-level parameterization strategy was able to outperform the single level method through a reduction in the number of open design variables. A number of the best solutions found are derived from the lowest level of parameterization and all contain more than one resonator tank within the bandpass filter system.

Comparing with the previous attempts at evolving bandpass filter transmission shapes the tri level example significantly underperforms, however as with the previous system – device level design problem all of the strategies outlined show little convergence and further exploration through the increase in functional evaluations may improve their performance.

8.3 Summary

This chapter introduced the final and most complex hierarchical MEMS design problem, consisting of system, device and physical level components. This brought with it a number of challenges, from the addition of an extra cost objective through the need to lower the von Mises

stress of the filter system, to the increase in the number of design variables open to the optimizers. The outcome of both single and multi-level design optimisation strategies were mixed with a significant degrading in performance in terms of bandpass filter synthesis when mirrored against system and system – device level results. However in relative terms the multi-level design strategies, particularly multi-level evaluation proved superior in comparison to the standard single level MOEAs.

9

Discussion and Conclusions

The final chapter draws together the research undertaken throughout this thesis for both multi-level and multidisciplinary design optimisation of MEMS providing a number of observations and conclusions on their application to MEMS design synthesis and optimisation. This is followed with a number of recommendations and future work linked to this piece of research that can be undertaken to further explore the role both multi-level and multidisciplinary optimisation have in MEMS design.

9.1 Discussion

The research reported in this thesis focused upon the automated design synthesis and optimisation of microelectromechanical systems with the introduction of a number of novel multi-level and multidisciplinary formulations. A critical review of the literature presented our research motivation, objectives and a thorough evaluation of the current state of the art in automated MEMS design in chapters 1 and 2. The evaluation of the current state of the art in multi-level and multidisciplinary optimisation was explored in chapter 3 and a number of novel strategies formulated for the application towards automated MEMS design synthesis and optimisation. A computational framework to house both the state of the art and proposed multi-level and multidisciplinary optimisation algorithms were outlined in chapter 4 and a hierarchical MEMS bandpass case study to allow evaluation of the proposed algorithms constructed and validated in chapter 5. Finally a series of MEMS design problems that spanned the various modelling and analysis levels of hierarchical design were constructed and used to evaluate each of the proposed multi-level and multidisciplinary optimisation strategies in chapter 6, 7 and 8.

A number of observations were made and outcomes established throughout the research in this thesis and the majority of these are drawn together here.

Automated MEMS design has grown over the last decade with stochastic methods such as those found within evolutionary computation beginning to establish themselves as the dominant search algorithm in terms of performance. The multi-objective algorithms MOGAII and NSGAII were applied to a number of MEMS design problems within the literature in order to build a platform for and update the field of automated MEMS design.

- [Chapter 2] The attraction of multi-objective evolutionary algorithms for application to automated MEMS design synthesis and optimisation comes from in part their stochastic

and population based nature which allows for problems with complex multi-modal, discontinuous and non-linear search spaces to be solved more effectively than with traditional approaches. The application of multi-objective evolutionary algorithms to a number of MEMS design problems yielded very good results, matching previous performance with a significant reduction in functional evaluations required, while also reducing the cost of the final designs, in this case total area of an ADXL150 accelerometer and micro gyroscope.

- [Section 2.4.5] Two of the most commonly used multi-objective genetic algorithms in MEMS design, NSGAII and MOGAII, were applied to each of the MEMS design problems with one outcome to investigate the performance between them. The performance of MOGAII in this instance proved superior for the set of experiments undertaken with better results in four out of the five case studies, and often producing a larger number of Pareto optimal solutions.

The application of multi-level and multidisciplinary optimisation strategies required the use of a MEMS benchmark problem that spanned a number of the hierarchical levels of modelling and analysis. The creation of a MEMS bandpass case study that contained system, device and physical level modelling and analysis was undertaken and validated over a range of filter frequencies.

- [Section 5.2.3] The electrical to mechanical conversion method outlined proved successful over a number of frequency ranges with either good or reasonable bandpass filter transmissions evolved over the design process. The design of the folded flexure resonator layouts associated with each of the bandpass filters also proved successful with each constituent resonator tank evolved to match design targets within 0.1% error.

- [Section 5.3] The design parameters for the electrical equivalent bandpass model heavily influence the starting position of solutions within the design search space with regards to the overall filter transmission shapes produced during initialisation. Altering the upper and lower bounds of the capacitance and inductance variables, in relation to the bandpass model causes the central frequency of the randomly initialised solutions to vary. The result of which causes degradation in performance as more time is spent by the optimizer evolving solutions closer to the target frequency range. Therefore careful consideration should be given when deciding the best parameters ranges for each variable in association with the bandpass filter targets set out by the designer.

- [Section 5.3] The overall design process of the GAECM methodology required 10,000 functional evaluations each to evolve both the filter transmissions and later their associated folded flexure resonator device layouts. In comparison with the current state of the art this is a significant improvement, with past attempts requiring 2.6 million and 137,500 functional evaluations to evolve both filter transmission and device layout design for comparable filter transmission shapes, an increase in performance of 260x and 14x respectively.

- [Section 7.1] The original GAECM methodology outlined in chapter 5 looked to evolve bandpass filter transmission designs at the system level and extract information and design targets for evolving the physical layout later at the device level of modelling and analysis. Another method was proposed which 'coupled' both the system and device level in a single design process with the physical layout designed and then analysed at the system level using its electrical equivalent model. The benefits of such an approach are that the system level response can be physically realized as it is based on an already evolved device layout design. Evolving bandpass filter transmission designs at the system level separately may give rise to design targets that cannot be realized through fabrication. Compared to the original strategy such an approach yielded final population sets which showed lower convergence to a Pareto front, the hypervolume performance also indicated no level of convergence with a positive linear trend towards improved performance as a result. The phenotypes of the evolved designs were also markedly different to the system level approach, showing flatter pass bands but lower stop band performance.

- [Section 7.1.1] Evolving the physical layout of the bandpass filter and the associated design variables of the folded flexure resonators within, compared with the electrical equivalent model design variables also gave rise to characteristic changes in optimization performance. With a much larger set of design variables, the system – device approach is open to more variation through the optimizer which transfers to a much larger change in phenotype from one offspring solution to the next.

- [Section 7.1.1] The conversion of the physical layout model and its associated mass and stiffness kx values into an electrical equivalent circuit model is one possible explanation for the difference in evolved phenotypes and overall performance. In the original system level formulation the design chooses the bounds on the capacitance and inductance variables that are associated with each tank of the electrical equivalent circuit model. While at the system – device level formulation the bounds and overall values of the capacitance and inductance can vary more greatly as they are associated directly with the mechanical values of the bandpass filter. It has been shown that this leads to both the system and system – device level design variables occupying different areas of the design search space as a result.

The hierarchical benchmark case study outlined and validated in chapter 5, a series of further design problems were than formulated from it, ranging in the number of modelling and analysis levels present and the overall complexity required to evolve solutions to the design problems. All of the optimisation strategies were than tested upon these design problems, and a comparison of performance between the current state of the art and the proposed multi-level strategies discussed. A number of observations were made over the course of experimentation for each of the strategies employed.

Single level optimisation

- [Chapters 6, 7 and 8] Overall the weakest of the strategies in terms of performance, the current state of the art single level or all-in-one MOEA approach showed characteristics

one might associate with the class of algorithm, a robustness of design, and ability to provide good solutions in what were often complex multi-modal design landscapes. However frailties in the overall optimisation structure or barriers associated with certain design problems retarded performance on certain occasions.

- [Section 6.3.1] The application of the single level strategy to the system level design problem often gave rise to the problem of convergence, particularly to a sub-optimal Pareto front. This proved problematic as small changes in the phenotype often meant large changes to objective values making it harder to vary solutions to more optimal regions of the search space.

- [Section 6.4.1] The effect of optimising device level folded flexure resonator devices to match design targets is affected greatly by the objective space used. The algorithm SPEA2 for example failed on both the single level and multi-level strategies to evolve solutions which contained an error of less than 1% from the synthesis targets outlined. This is in part due to the use of three objectives, with the cost objective being an antagonistic force to the detriment of the other synthesis objectives. However probably the most significant factor is the effect of the distance metric employed within the SPEA2 algorithm. The loss of high fitness individuals occurs throughout the design process due to the objective values present and the distance between neighbouring solutions being much closer between the two synthesis objectives than when paired against the total area objective.

- [Section 6.4.8] Normalizing the objectives within the SPEA2 distance metric calculation proved a significant boost to performance, finding solutions which had an error less than 1% from the design synthesis targets outlined and greatly improving the average synthesis error of the whole population throughout the design process.

- [Section 7.1.1] Reversing the process for bandpass filter design to evolve the 2D layout first and then analyse the function using an electrical equivalent model in what is a system – device level design problem showed promising results compared with the standard approach. Evolving the layout and the constituent folded flexure resonator models seems to have slowed the level of convergence and as a result lowered the chance of the optimizer getting stuck within a sub-optimal region of the design search space. There is also a change in the phenotype of the evolved bandpass filter transmission shapes with a broader, flatter pass band but a shallower stop band.

- [Section 7.1.1] A large part of this lack of convergence and ability to possibly escape sub-optimal regions stems from the larger number of design variables present within the representations used to model the bandpass filter. The mechanical version is open to more variation than the electrical equivalent model which may aid the optimizer in evolving a more diverse set of solutions over the course of the design process.

- [Section 7.2.1] The limited availability of computational resources when undergoing design optimisation that involves complex FEA/BEA models can lead to poor

performance of the single level optimizers, particularly when trying to optimise a number of synthesis objectives. The optimisation of the physical – device level design problem was unable to evolve solutions which matched the targets set out for the coupling spring component.

- [Section 8.1.4] The design optimisation of a bandpass filter using all three levels of modelling and analysis proved unsuccessful, with very poor filter transmissions being evolved over both NSGAII and SPEA2 algorithms. The large number of design variables associated overall with the bandpass filter model and the low number of functional evaluations available to the optimizers proved too much of a challenge. The addition of a third cost objective may have also played a role in degrading performance with the optimizer having to focus upon this extra objective.

Multi-level evaluation

- [Chapters 6, 7 and 8] Showed over the course of experimentation the largest boost to performance and design, often as a result of the additional functional evaluations made available through the strategy.

- [Section 6.3.3] The system level design of a bandpass filter was greatly enhanced using a multi-level evaluation strategy with performance matching the current state of the art single level strategy at around a fifth of the computational cost. Further evolution pushed designs to have superior bandpass filter transmissions over the ten experimental runs of NSGAII and SPEA2. The design search for the system level problem portrayed issues of discontinuity which might explain why the single level approach stalled at sub-optimal regions while the multi-level evaluation strategy was able to bridge the gaps and find more optimal regions of the search space.

- [Section 6.3.3] The issue of convergence to sub-optimal regions of the design search space were abated with a reduction of around 70% of experimental runs converging to a sub-optimal region to around 10% of runs for both NSGAII and SPEA2. An investigation into the role migration has between multiple population sets showed how lower level solutions successfully transfer better solutions to the highest levels early on in the design process.

- [Section 6.3.8] Multi deme strategies involve the use of more than one population during the design process, in this instance for each level of a multi-level strategy. The approach in utilising multiple populations may inherently be the driver for the improved performance seen within the multi-level strategies employed. A repeat of the system multi-level evaluation experimentation using only a single population however produced similar performance to that of the multi deme strategy. There was however an indication that the changes to the objective space through the use of low accuracy modelling could provide some performance ‘boost’ early on in the design process for the system level problem.

- [Section 6.4.3] Coupling together a low cost analytical model with a high cost NODAL model of a folded flexure resonator brings with it both the increased number of functional evaluations made available to the optimizer, vastly improving the design process but also an ancillary benefit with the analytical model holding structural design knowledge through its rigid topology. This simple topology allowed for much smaller total areas to be evolved compared with using only the higher cost NODAL model at the single level strategy. The addition of the NODAL model at a higher level also allowed some fluidity in design from the optimised rigid models found using the analytical model, which could now be improved further.

- [Section 6.5.3] Applying a strategy of multiple levels of FEA meshing in order to increase the number of functional evaluations and as a direct result improve design performance was limited with the spreading of functional evaluations over two separate populations not sufficient enough to counter the loss of convergence within these populations.

- [Section 7.1.3] Migration between multiple levels aids the design process within the system – device level design problem, with solutions evolved at the lowest level of modelling and analysis able to migrate upwards and seed the higher levels with improved designs. Migration of solutions is more successful going from a lower level of modelling and analysis than from a higher level for this set of design problems.

- [Section 7.2.2] The availability of multiple modelling and analysis tools across more than one level of the hierarchy of MEMS design can provide a much larger computational cost saving then would otherwise be possible using only one level of modelling and analysis tools. The coupling of a device level NODAL model with a physical level FEA model provided a computational saving of 50x that of the standard FEA model meshing approach used within the physical level design problem of chapter 6. The downside is that there is a trade-off in the accuracy of evaluation with the NODAL model showing a larger discrepancy then the simpler meshing approach.

- [Section 7.2.3] Opening up the physical – device level design problem to a longer design process through the additional availability of more functional evaluations has had a profound effect on the overall optimisation performance. The approach was able to produce a 94% decrease in the synthesis error of the coupling spring components evolved compared to those found using the single level strategy. The overall performance saw the multi-level evaluation strategy able to reach performance levels of the single level strategy at a fraction of the computational cost requiring only a fifth of the cost.

- [Section 8.1.4] The increase in the number of functional evaluations open for design allowed the multi-level evaluation strategy to post the best set of results for the system – device – physical level design problem. Once again the strategy is able to match the performance of the single level approach while only utilising a fifth of the computational cost to do so.

Multi-level parameterization

- [Chapters 6, 7 and 8] Mixed performance, often related to the design problem at hand, however in comparison to the standard single level strategy it is comparable or superior in performance.

- [Section 6.3.5] The addition of multiple populations, each with multiple levels of representation lead to the formation of separate Pareto fronts at either level on a number of experiments using the multi-level parameterization strategy. A clear indication that such a strategy alters the design search space by its effect on what variables are open to variation and what degree of variation is available through the parameter bounds placed upon them.

- [Section 6.3.5] The multi-level parameterization approach showed improved performance over the single level strategy, mainly through solutions of the highest and unrestricted level of parameterization. The characteristics of the overall strategy and design process were however not always uniform with some runs exhibiting levels which converged to a front while other levels were able to break free, another run showed each level mirroring one another with population members seemingly able to migrate freely, while another example saw lower level Pareto fronts shifting to greater performance at the higher full parameterization level.

- [Section 6.4.5] The lowest level of parameterization for the folded flexure resonator of the device level problem provided early rapid gains in optimisation performance with the more complex levels of parameterization catching up and overtaking later on in the design process. In the end the best solutions are found at the highest level of parameterization.

- [Section 6.4.5] Each of the levels specific representations are able to aid the design process with lower level solutions migrating up throughout the design process and along the way being optimised and evolved to match the design targets set out.

- [Section 6.5.5] The application of a multi-level parameterization scheme to the physical level coupling spring design problem when successful was able to provide more Pareto solutions with a synthesis error of less than 1%, however this was not often enough with a number of runs failing to find any solutions. The step wise progression of improvement of solutions migrating between the separate levels of parameterization was once again observed.

- [Section 7.1.5] A multi-level parameterization strategy can also produce negative effects upon the design process with solutions evolved at the lowest level of parameterization migrating to higher levels and forcing early convergence of the population. This early convergence could be towards sub-optimal designs which the optimizer can fail to break free from. A migration policy which is invoked often and early in the design process as

seen at the system – device multi-level parameterization problem may be a root cause to this.

- [Section 8.1.4] The simplified parameterization of a system – device – physical design problem representation allowed for a number of superior bandpass filter designs to be evolved over the design process in comparison with the single level strategy, with a large percentage of the final best designs coming from the lowest level of parameterization.

Multidisciplinary optimisation

- [Chapters 6, 7 and 8] A novel population-based multidisciplinary optimisation algorithm applied to a number of design problems through an object-based decomposition strategy with the goal to simplify and improve the design optimisation process.

- [Section 6.3.7] The application of MDO to the system level design problem provided comparable performance against the single level SPEA2, while superior performance in relation to NSGAII. The structured nature of MDO with separate subsystems optimising different population sets containing different objective and constraints can lower the level of population convergence, less so for NSGAII, while also adding an additional cost through the loss of functional evaluations during re-analysis at the subsystem and system levels.

- [Section 6.3.7] Population members moving from system to subsystem or vice versa can have their performance lowered as a result of different objective spaces present within either level. This can lead to wasted search as the subsystem has to ‘re-evolve’ solutions to fill its local search space, in this instance a result of the new objectives and constraints present within the subsystem which can differ to those of the system level. Care therefore needs to be given so as to reduce the division between upper and lower level objectives so as to smooth out or reduce any loss of performance as they migrate from each level.

- [Section 6.3.7] Decomposing the system level problem into a number of subsystems, each with their own objectives and constraints has shown the ability to focus solutions to match those local objectives, whether it is to smooth out the pass band of the filter transmission or increase the bandwidth. The decomposition also plays a role in the genotypes of the solutions held within each subsystem. The correlation between the separate subsystems is not always equal with different ‘allele’s present in a number of variables within each subsystem. The local fitness associated with theses alleles can also vary with certain alleles having a higher local payoff than in other subsystems. There is also a deleterious effect on certain alleles shown throughout the design process as a result of when the system level offspring ‘updates’ each subsystem. Once strong alleles with a large presence within the subsystem population and containing high local objective payoff are removed and replaced with different values, no doubt influenced by the system level.

- [Section 6.4.7] Multidisciplinary optimisation of the device level folded flexure resonator problem through object decomposition led to a characteristic change in both NSGAII and SPEA2 algorithms with a focus placed upon the synthesis objectives and less domination of the optimizer to evolve designs with very small total areas. This gave rise to a larger percentage of solutions which lay within 1% error of the design synthesis targets compared with the single level strategy.

- [Section 6.4.7] The MDO strategy is able to drive the synthesis values of the best solutions and overall average population values too much lower levels of error than found using the single level strategy which often stalls at prohibitively high synthesis error values. Each subsystem was able to evolve their targeted components of the central mass or folded flexure springs and then later through recombination of the 'alleles' associated with them produce fully optimal solutions.

- [Section 6.5.7] Assigning control of specific design variables to a subsystem can have adverse effects on the performance of other subsystems and their internal objectives. The application of MDO to the physical level problem resulted in much higher von Mises stress within the evolved solution on average over the design process. This is a result of one subsystem have a dominant control over the sizing of the coupling spring that was to be evolved and flooding the other subsystem with alleles for a coupling spring shape which contained high von Mises stress values. With no control over this particular design variable the subsystem had little ability to evolve solutions which contained much lower von Mises stress levels.

- [Section 6.5.7] The MDO strategy was once again able to evolve and recombine separate functional units of a whole device into a more optimal design and do so throughout the design process.

- [Section 7.2.5] Solving the physical – device level design problem through the decomposition into a number of separate subsystems with each focusing on a set of specific objectives and constraints proved highly successful when compared with the single level methodology. The level of synthesis error within both the folded flexure resonator and coupling spring are significantly reduced when compared with the single level approach.

- [Section 7.2.5] The MDO strategy is able to produce a structured design process which allows local optimisation to occur within each of the specific subsystems while also allowing good 'alleles' to be transferred between each of these subsystems and as a result evolve solutions which are both locally optimal but also contain a relatively high global fitness with regards to the other subsystems and their local objectives. The result of such transfer of good building blocks is the improvement of a solution which contained a synthesis error in the range of 100K% to the final solution containing only 2% error.

9.2 Contributions

The major contributions of this work are:

- *Multi-level Design Optimisation of MEMS:* A comprehensive review of the literature provided the backdrop for outlining two unique multi-level design optimisation strategies for MEMS design. The developed multi-level evaluation and parameterization strategies have been tested and compared with the current state of the art in automated MEMS design and show substantial improvements to performance of the overall design process. The contributions each strategy brings to the design process is also discussed and analysed, with a focus upon their application to MEMS design problems, particularly what type of problem they should be applied and where each strategy can provide the best outcome for design optimisation.

- *Multidisciplinary Optimisation of MEMS:* A comprehensive review of the literature of multidisciplinary optimisation and in particular population-based methods was followed with the construction of a novel population-based MDO algorithm for MEMS design. The algorithm is designed to overcome limitations in previous state of the art MDO algorithms present within the literature and open it up for application to the field of MEMS. An object-based decomposition of a number of MEMS design problems provided the ability for testing and comparing the MDO strategy against the current state of the art in automated MEMS design. Results show significant improvement on a number of design problem examples and is followed up with analysis and discussion of how best to utilise the field of MDO in MEMS design.

- *Computational Modular Framework:* The construction of a computational modular framework tailored towards design optimisation and the field of evolutionary computation. The framework is uniquely designed to handle the fields of evolutionary computation, multi-objective, multi-level and multidisciplinary optimisation, and engineering design through integration with a host of modelling and analysis tools. A dynamic, tree-based representation is utilised throughout the MEMS design problem experimentation and is aided with a number of ‘decorations’ to allow further detailed control required for multi-level and multidisciplinary optimisation.

- *Hierarchical MEMS Benchmark Case Study:* A series of hierarchical MEMS design problems covering three levels of modelling and analysis (System, Device and Physical). The benchmark is built from six separate design problems associated with the design synthesis and optimisation of a MEMS bandpass filter. Each of these design problems have been used to create a series of novel multi-level and multidisciplinary optimisation formulations focusing on a single or series of coupled modelling and analysis levels.

- *Automated Stochastic MEMS Design Optimisation:* A series of MEMS design problems are taken from the literature and used to evaluate some of the current state of the art multi-objective design optimisation algorithms. Within the MEMS design literature two multi-objective evolutionary algorithms are referenced most, MOGA [44] and NSGAII

[45]. A series of experiments explore and compare both algorithms performance to MEMS design updating the literature with some of the more current state of the art results.

9.3 Limitations

The application of multi-level and multidisciplinary optimisation to the field of MEMS covers a wide range of disciplines and a large body of research. As a result there are a number of constraints applicable to this research which needs to be highlighted and addressed.

- The hierarchical design process within MEMS provides designer with access to a large number of modelling and analysis tools with varying levels of granularity and computational cost. However in order to undertake automated MEMS design synthesis and optimisation, these tools require some level of interface which allows for their modelling and analysis function to be called outside of the tools themselves, often through the use of some kind of application programming interface (API). Unfortunately not every tool has such an interface and therefore it is impossible to hook the modelling and analysis tools into an automated design optimisation platform. This restricts what tools are available and what modelling representations can be used when looking to construct a suitable hierarchical case study benchmark.

- Multi-level design optimisation as a field of research is slightly ambiguous which as a result leads to a large number of strategies and cross talk into other fields that exhibit multi-level behaviour. Deconstructing it to its basic premise, multi-level design simply outlines design strategies which contain multiple entities that are in some way partitioned into separate levels or sets based upon some criteria. A simple example presented in [46] presents a multi-level strategy as two separate levels each containing their own optimization routines, in this instance a global search optimizer at one level and a local search optimizer at the other. Solutions can be created, evolved and migrated between each level over the design process in a routine similar to a separate thread of research that of hybrid optimisation. Hybrid optimisation often involves the coupling of a global search algorithm such as an evolutionary algorithm to a local gradient based search algorithm with the aim of exploiting both global and local search. Depending on their construction such an approach can be seen as multi-level, often involving communication between each level over the entire design process, or as a hybrid algorithm, often with local search simply performed at the end on the best solutions found by the global optimizer. The salient point is that it is not possible to undertake all possible multi-level strategies available and careful consideration has to be given on what can be classed as a true multi-level design strategy and another which simply exhibits some of its features.

- This body of research at the highest level can be split into two distinct areas, firstly that of computer science and the application of soft computing techniques to the automated design of MEMS, and secondly the industry of microelectromechanical systems itself. Though it is reasonably possible to validate those contributions within the thesis arising

from the application of the soft computing techniques, when it comes to validation of the MEMS devices created, in particular through fabrication, it is more difficult. Unfortunately the resources are not available to fabricate any of the novel MEMS devices created throughout the thesis and therefore special care has to be taken when firstly evaluating the role of multi-objective genetic algorithms in MEMS design and then later when creating the hierarchical MEMS case study benchmark design problems. Therefore wherever possible the MEMS devices used within this thesis were taken from the literature, from fabricated and realisable designs, which have themselves played a part in automated MEMS design research.

9.4 Future Research

The application of multi-level and multidisciplinary optimisation to MEMS design synthesis has been explored with this research providing a number of observations and conclusions. The following section now looks to expand upon some of these observations by presenting a number of new directions for research and where applicable further development of work undertaken.

- **Meta models**

The multi-level evaluation strategy outlined in this thesis exploits the hierarchy of modelling and analysis tools available to MEMS designers to speed up the process of design. An alternative to using separate modelling tools is to incorporate routines which build low cost surrogates or meta models often trained dynamically from previously evaluated individuals [205]. Meta model evolutionary algorithms have been discussed and researched on a number of occasions [34][46][182][205] and it would be advantageous to investigate whether it is simply better to use MEMS specific tools, meta models or a mixture of both when it comes to MEMS design.

- **Self Healing / Robust Design**

Manufacturing and environmental variability are an important factor when it comes to MEMS design and their eventual fabrication and function. In an ideal world the final realized device would match the specifications of the various mask layouts used in its fabrication to produce a device that mirrors the designers 'on paper' or *in silica* counterpart. Unfortunately fabrication error or variability can lead to designs which contain a level of mismatch that can alter the final function or behaviour of the device from its original intended design. Even with a perfectly realized device there is still the problem of environmental instability and variability, for example temperature, which can also lead to changes in how the device will operate.

There are two relatively new paths of research which look to overcome some of these difficulties, the field of 'self healing' and 'robust design'. Self healing microsystems look to overcome the influences of the environment or intrinsic aging through the design of systems that can autonomously select components or set values to achieve some level of resilience [298]. An example is shown in figure 9.1 where an array of micro resonator devices are used to build a MEMS bandpass filter device with individual resonators that can be dynamically tuned on or off to match a target filter transmission shape.

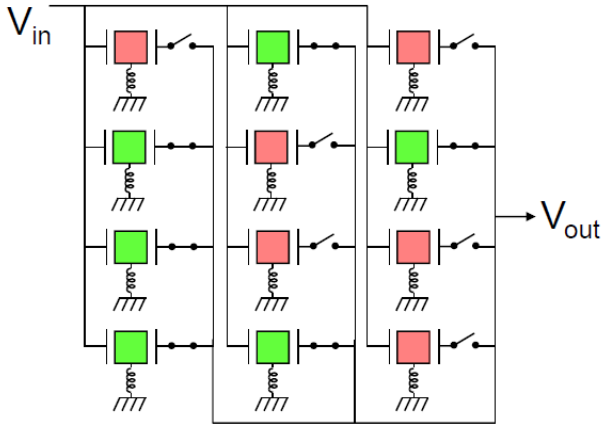


Figure 9.1 Self healing RF micro resonator array [298]

A similar method in robust design optimisation also looks to build resilience into designs against manufacturing variability through devices or components whose function is tolerant to change, be it environmental or physical. Both examples incur a large computational expense, whether it is from repeated analysis as a result of Monte Carlo variation to model uncertainty or the need to analysis large numbers of components as a result of redundancy in self healing systems. It may be beneficial to incorporate a multi-level evaluation routine into areas such as MEMS robust or self healing design optimisation to speed up the design process.

- **Single Deme v Multi Deme**

Investigated previously within this thesis was to what effect the numbers of populations or demes present within the optimisation routine had on the performance of the multi-level strategies outlined. In the design optimisation of a system level bandpass filter both a single deme and multi deme multi-level evaluation strategy was employed and compared. This small set of results pointed to similar levels of performance regarding whether to use a single or multiple deme strategy, however there were characteristic changes to performance early in the design process. In the past there has been research into both single deme and multi deme multi-level strategies but a conclusive investigation into which approach is better or perhaps which is more applicable to certain design problems has not been investigated. Having constructed a large body of work already within this thesis centred on multi deme multi-level optimisation it would be prudent to use this work and continue it with an investigation into single deme multi-level optimisation as well.

- **Process level design optimisation**

The scope of this thesis and research focused primarily on the hierarchical system, device and physical levels of the MEMS design process. However there is one more level applicable to the design process and that is the level which handles fabrication and process information to build the MEMS device itself. This ‘Process’ level is not often used for automated design optimisation given the difficulty in optimising process and mask layout information, and how these changes are permeated through to the final functional device. However the need to

produce such process and mask layout information for the final optimised device warrants further investigation into how to incorporate the process level of design.

- **Multi-level representation strategy**

This thesis through its literature review on multi-level design optimisation outlined three main strategies employed throughout the literature, characterised as 'multi-level evaluation', 'multi-level parameterization' and 'multi-level search'. Each look to exploit some characteristic of the design process be it multiple levels of evaluation software or different methods of global or local optimisation. A new multi-level strategy similar in some way to multi-level parameterization looks to exploit the different levels of problem representation available within the literature. A clear example can be seen here in this thesis and in past work undertaken by Fan [138] who both looked to optimise a series of MEMS bandpass filters however utilising different representations. The use of a tree based GP representation found in [52][138] brings with it additional benefits to topological search and the possibility of more creative designs. The use of a more traditional array of real-valued design variable associated with the typical evolutionary algorithm and seen throughout this thesis may also provide a more concentrated local search on specific design variables once a topology is set. The combination of both representations within a multi-level representation strategy could prove beneficial to the design process and is worthy of investigation.

9.5 Conclusions

This thesis has presented a successful study into the role multi-level and multidisciplinary design optimisation strategies can play in the field of MEMS design synthesis and optimisation. It was argued that the current field of MEMS design optimisation does not fully exploit the hierarchical nature of MEMS synthesis associated with the modelling and analysis tools presented to a designer. It was postulated that hierarchical strategies from the field of multi-level and multidisciplinary design optimisation would be able to exploit the hierarchical nature of MEMS design and help the design process of MEMS. To support this notion this thesis has devised, tested and analysed several new multi-level and multidisciplinary optimisation strategies on a number of MEMS design problems and argued for their contribution to automated MEMS design.

A discussion on the wider status of traditional MEMS design synthesis and optimisation before focusing on automated and non-traditional approaches was presented through a literature review in chapter 2. This included a breakdown of the modelling and analysis tools present within MEMS design and an argument for the use of more stochastic, non traditional optimisation algorithms for automated MEMS synthesis. An argument for evolutionary computation, particularly multi-objective genetic algorithms as the current state of the art in automated MEMS design optimisation was put forward. This was followed with a series of experiments to evaluate and validate their application to MEMS design synthesis and optimisation and additionally set out the schema for connecting and communicating between the optimisation platform and modelling and analysis tools used throughout this thesis.

The next element of the thesis began with an exploration into the field of multi-level and multidisciplinary design optimisation strategies with a focus upon their application to

engineering design optimisation problems. From this a number of strategies applicable to MEMS design were extracted and outlined in chapter 3, concerning how best to overcome the high computational cost of design present within automated MEMS synthesis, the complexity of the design search landscape and the highly multidisciplinary nature of MEMS. Having established multi-objective genetic algorithms as the current state of the art in automated MEMS design and outlined a number of suitable multi-level and multidisciplinary optimisation strategies for application to MEMS design synthesis, chapter 4 was tasked with presenting the modular design optimisation framework built to house and deploy each strategy. The modular framework was also used to build the base multi-objective genetic algorithms (NSGAII and SPEA2) used throughout the remainder of the thesis, and along with a novel non-hierarchical MDO algorithm were validated against a number of design problems. Both sets of algorithms matched comparatively with the performance of the previous state of the art results within the literature and with regards to the MDO algorithm at a fraction of the computational cost.

Chapter 5 constructed a hierarchical MEMS bandpass filter case study and followed up with a set of empirical studies into the design optimisation of a series of filter design problems. The new approach coined 'GAECM' showed an ability to evolve designs which matched the target specifications; however with depreciation in quality as the frequency range was increased. This was shown to be a result of the starting parameters of the design problem in relation to where the optimisation starts within the design search space. Overall however both the evolution of circuit level bandpass filter models and their realized 2D mechanical filter designs showed a matched performance with the current state of the art [53] at a significant reduction in computational cost.

The final section runs through chapters 6, 7 and 8, constructing a series of MEMS hierarchical design problems and using them for a number of empirical studies into each of the multi-level and multidisciplinary optimisation strategies outlined in chapters 3 and 4. The results of which were compared against the current state of the art in automated MEMS design optimisation, multi-objective evolutionary algorithms. It was shown that the application of multi-level and multidisciplinary optimisation strategies can be applied as both an 'Intra-level' methodology utilizing only a single modelling and analysis tool or as an 'Inter-level' methodology using multiple levels of modelling and analysis tools for MEMS optimisation. From the set of experiments a number of observations were made covering both benefits and drawbacks of the application of the multi-level and multidisciplinary optimisation strategies outlined. These include an increase in performance through the application of a multi-level evaluation strategy and the additional functional evaluations made available as a result of exploiting the use of less computationally expensive analysis. A decrease in performance was also observed due to the use of multiple populations which in some instances lowered convergence to good solutions and wasted search effort as functional evaluations were lost revaluating migrating solutions.

Applying a multi-level parameterization strategy showed an ability to reduce the complexity of the design search space and optimisation routine and as a result aid the overall design process. Reducing the number of active design variables allowed more simplified and structured devices to be evolved which could then by migrated upwards to levels of design with more freedom. This allowed designs that matched near optimal design synthesis targets to be evolved quickly and then later improved further at higher levels of design. A novel population based non-hierarchical

multidisciplinary optimisation algorithm developed within this thesis showed remarkable success on a number of design problems along with a much muted performance on others. It was shown that it is possible to decompose individual MEMS devices into a number of objects and optimise them individually before recombining them into a whole, fully optimised device. Equally the MDO algorithm was suited to evolving and optimising much larger systems composing of multiple components / devices all cooperating together to form a single function.

Micro-electro-mechanical systems are a field grown out of the integrated circuit industry, utilizing fabrication techniques from the technology of VLSI and with the goal of developing smart micro devices which can interact with the environment in some form. They promise to revolutionise our present day lifestyles as much as the integrated circuit has done in recent decades. In order to push forward this revolution a dedication towards improving the design process and opening it up to application designers is needed. Over the last decade work has been undertaken to improve the process of automated MEMS design through the building of more advanced and accurate modelling and analysis tools, along with an integration of advanced optimisation routines. This has seen the inclusion of stochastic multi-objective evolutionary algorithms to aid automated design over the more traditional gradient based approaches, integrating interactive methods to allow designers to mould and direct the design optimisation process and linking the process with case-based reasoning methodologies to act as an automated knowledge base for selecting promising structures. This thesis has looked to once again extend this boundary of knowledge and improve the MEMS design process through the application of multi-level and multidisciplinary optimisation strategies.

A

Appendix

Multi-objective design optimisation of MEMS

A.1 **Design Optimisation Platform**

In order to undertake a successful validation of multi-objective evolutionary algorithms for MEMS design optimisation, an optimisation platform and framework for linking the various modelling and analysis tools is needed. The hierarchical and multi-disciplinary nature of MEMS gives rise to a number of modelling tools, each of which may need to be interfaced with the optimisation routine and therefore routines for accessing and manipulating these tools through an optimisation platform is ideal. The commercial software platform modeFrontier [145][146] is one such ideal tool, marrying an optimisation platform consisting of a number of state of the art algorithms with a host of commonly used multidisciplinary modelling and analysis tools used throughout the literature and industry [51][87][89]. The optimisation platform contains a template for integrating any number of computer aided design tools so long as they contain necessary application programming interfaces (APIs). The generic framework is outlined in figure A.1 and the platform workspace is shown in figure A.2.

The process for design optimisation begins as with most engineering design problems as a simple definition of the goals or objectives set out by the designer. These have to be tied in some way to a parameterised model and a set of design variables for the optimisation routine to act upon need to be defined. The design optimisation platform workspace allows for a number of individual input variables to be defined which hold information on the decision variables which the optimisation algorithm will act upon. These variables can either be passed to a file template module which both holds and overrides any user specified simulation model, or simply passed directly to the CAD/CAE simulator. A number of CAD/CAE simulator modules present in the platform allow a direct interface through the API or if none is present various batch scripting can be called instead to run specific simulations. Upon completions of any simulation event output results either through specific CAD/CAE output files or directly through their specific

workspaces can be extracted. A number of output variables which hold the desired information can now be used for objective calculation.

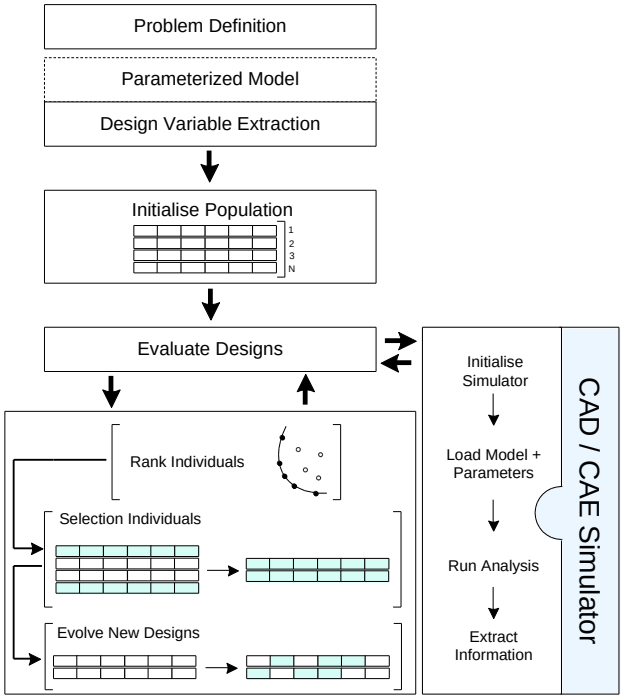


Figure A.1 Generic framework for multi-objective design optimisation

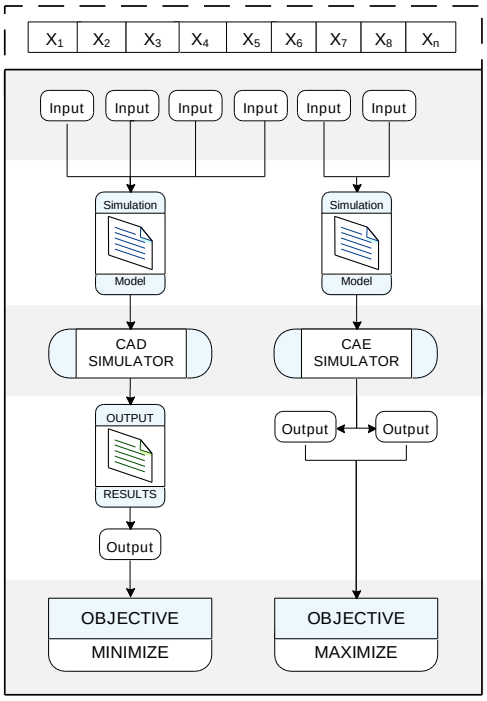


Figure A.2 Design optimisation platform workspace

Within the design optimisation platform are two state of the art multi-objective genetic algorithms, MOGAII [44] and NSGAII [45]. It is these two algorithms that have been chosen to evaluate the efficacy of multi-objective genetic algorithms to MEMS design optimisation.

MOGAII is an improved version of MOGA by Poles [44], utilizing a smart multi search elitism, and a triad of operators (classical one-point crossover, directional crossover and bit flip mutation) each with their own probability of invocation. Directional crossover looks to find a direction of improvement using two reference individuals when evolving a particular offspring. The two individuals, Ind_j and Ind_k, are assigned from a group of randomly chosen individuals based upon fitness. A new individual is then created by moving in a randomly weighted direction that lies within the sphere of influence of all three individuals as seen in figure A.3.

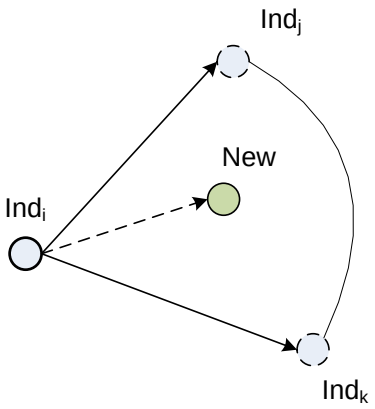


Figure A.3 MOGAII directional crossover

As with classical MOGA, the representation is a binary string and in order to simulate continuous variables a sufficiently high base value must be used to divide between upper and lower bounds the possible variable values, an outline of MOGAII is given in algorithm one. NSGAII [45] is an elite preserving multi objective genetic algorithm, which also includes a diversity heuristic to maintain a uniform spread on the Pareto front. Unlike the standard MOGA, NSGAII uses a real-valued representation, and therefore both recombination and mutation operators revolve around these real values, an outline of NSGAII is presented in algorithm two.

Algorithm 1: MOGAII Pseudo Code

1. Initialise population
    (a) Generate random population of size N and elite set $E = \theta$
2. Evaluate objective values
3. Assign rank based on Pareto dominance – 'Sort'
4. Generate offspring population
    (a) Combine both population and elite sets $P' = P \cup E$
    (b) If the cardinality of P' is greater than the cardinality of P reduce P' removing randomly the exceeding points
    (c) Compute the evolution from P' to P'' applying MOGA operators:
        i. Randomly assign one operator (local tournament selection, directional crossover, one-point crossover or bit flip mutation) based upon probability of invocation
5. Evaluate objective values of population P''
6. Assign rank to P'' individuals based on Pareto dominance – 'Sort'
7. Copy all non-dominated designs of P'' to E - 'Sort'
8. Update E by removing duplicated or dominated designs
9. Resize the elite set E if it is bigger than the generation size N removing randomly the exceeding individuals
10. Return to step 2 considering P'' as the new P until termination

Algorithm 2: NSGAII Pseudo Code

1. Initialise population
   (a) Generate random population of size N and elite set $E = \theta$
2. Evaluate objective values
3. Assign rank based on Pareto dominance – 'Fast-Sort'
4. Generate offspring population
   (a) Create population P' using tournament selection and apply variation operators (Simulated binary crossover and polynomial mutation)
5. Evaluate objective values of population P'
6. Combine both population sets P and P' to give set size of $2N$ P''
7. Assign rank to P'' individuals based on Pareto dominance – 'Fast-Sort'
   (a) Fill new P set with non-dominated fronts until cardinality is reached from set P''
   (b) If cardinality of new set P is greater than the size N reduce P by computing the crowding distance of the last front set to be added and fill remaining slots using crowded-comparison operator
8. Return to step 4 until termination

A.1.1 MEMS multi-objective design case studies

In order to validate the use of multi-objective genetic algorithms as the choice of algorithm for the design optimisation of MEMS within the thesis a set of MEMS design optimisation case studies taken and adapted from the literature are needed. Therefore a set of 5 case studies of increasing complexity and covering two different methods of modelling and analysis of MEMS are constructed using both nodal and analytical modelling.

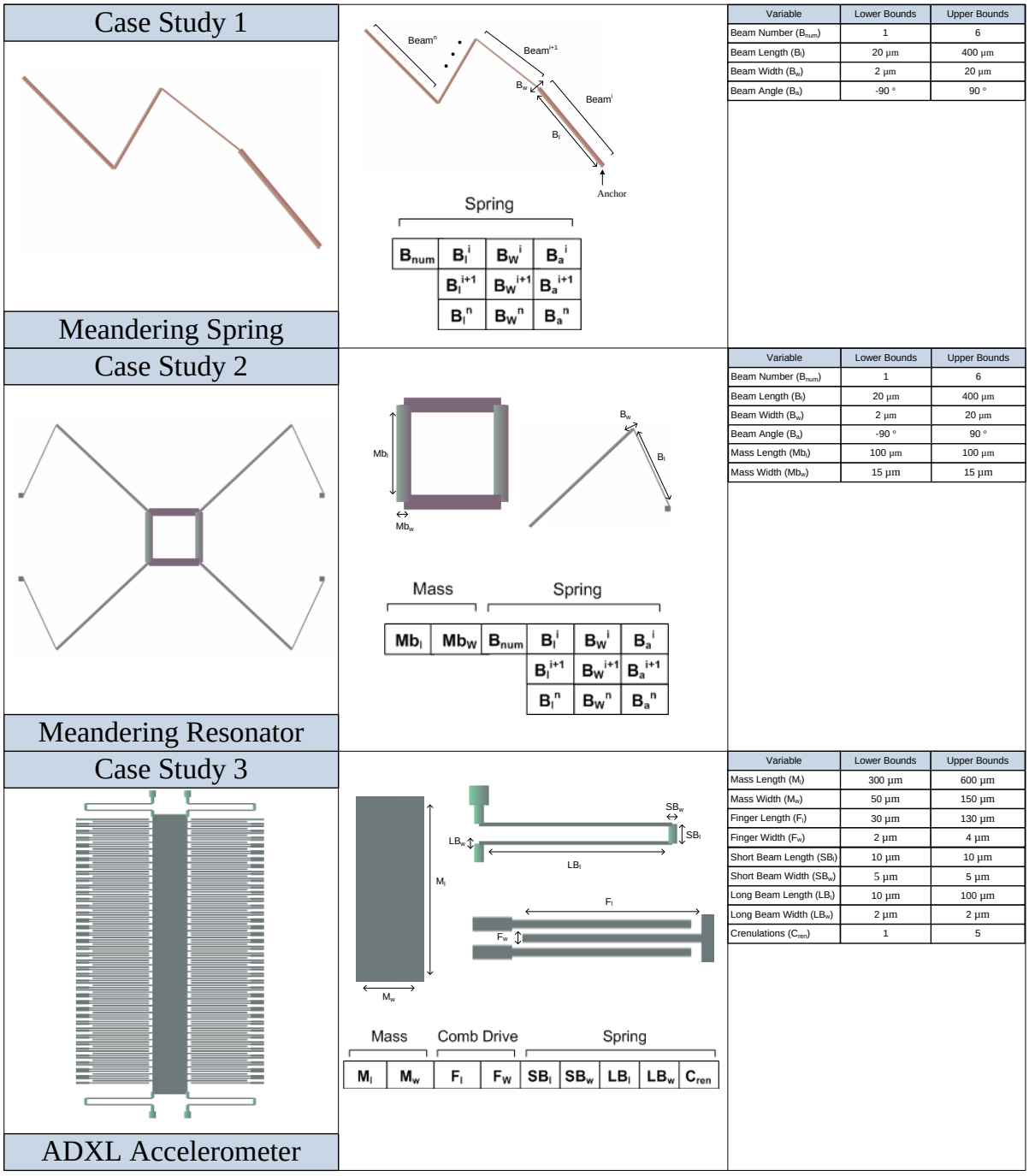
The integration of the Sugar [66] platform into the design optimisation framework allows for suitable 'device level' nodal modelling and analysis to be undertaken. This can be coupled with the proposed multi-objective algorithms, providing a foundation for the design optimisation of MEMS.

Extending previous work undertaken in the field [31][32][42][137], planar MEMS devices form the basis of our evaluation of our design optimisation approach and forms a suitable strategy to evaluate the performance of the considered algorithms. A series of case studies with increasing complexity (meandering spring, meandering resonator and ADXL150 accelerometer) are illustrated in figure A.4. All case studies are based upon poly-silicon material.

Case study 1 – Meandering spring

The core topology of a large class of MEMS, such as micro-resonators and accelerometers, consists generally of a spring + mass system, where a mass is suspended by a spring like structure anchored to a substrate. The shape and topology of the spring structure effects the behaviour of the device. Therefore, the ability to evolve spring like structures which match certain behaviour characteristics is important for the design optimisation of more complex spring and mass systems such as a micro-resonator. Following previous work [137] a simple meandering spring is synthesised and is composed of several beams, each of which has three variables (length, width and angle). The model design parameters are shown in figure A.4 and in this instance the representation consists of a mixed real-valued and integer variable length

chromosome. This has the ability to add or remove the number of beams linked in the whole spring structure.



Variable	Lower Bounds	Upper Bounds
Beam Number (B_{num})	1	6
Beam Length (B_l)	20 μm	400 μm
Beam Width (B_w)	2 μm	20 μm
Beam Angle (B_a)	-90 °	90 °

Variable	Lower Bounds	Upper Bounds
Beam Number (B_{num})	1	6
Beam Length (B_l)	20 μm	400 μm
Beam Width (B_w)	2 μm	20 μm
Beam Angle (B_a)	-90 °	90 °
Mass Length (Mb_l)	100 μm	100 μm
Mass Width (Mb_w)	15 μm	15 μm

Variable	Lower Bounds	Upper Bounds
Mass Length (M_l)	300 μm	600 μm
Mass Width (M_w)	50 μm	150 μm
Finger Length (F_l)	30 μm	130 μm
Finger Width (F_w)	2 μm	4 μm
Short Beam Length (SB_l)	10 μm	10 μm
Short Beam Width (SB_w)	5 μm	5 μm
Long Beam Length (LB_l)	10 μm	100 μm
Long Beam Width (LB_w)	2 μm	2 μm
Crenulations (C_{ren})	1	5

Figure A.4 Nodal case studies

Case study 2 – Meandering resonator

It is important to ensure that the design of a micro resonator matches a certain frequency performance, so that it can be integrated, for example, into a band-pass filter device [50][53]. Following previous work [32] a MEMS resonator is evolved in order to match certain behaviour and design objectives. For this case study a set of four meandering springs are evolved, each of which consists of several beams. Figure A.4 contains the model design variables for this device,

where a single spring contains the same set of variables as the previous case study. The mass is represented by two variables, length and width. In order to reduce the search space complexity, a symmetry constraint to the design is applied, where only one spring needs to be evolved and is then mirrored in both the x and y axes.

Case study 3 – ADXL150 accelerometer

MEMS technology offers the possibility of producing devices that mimic already viable real world macro designs but at a much smaller and more energy efficient scale. The ADXL accelerometer series is a device which has been fabricated and tested in real world applications and has replaced its macro-scale counterpart. This device can detect acceleration, as a result of experiencing forces imposed upon it (including gravity). It is a crucial component of many applications such as that of car airbag deployment. As a result of impact with another vehicle, the resulting vehicle acceleration is detected by the accelerometer and if the corresponding value exceeds a given threshold the airbag is deployed, saving lives. The design variables derived from [42] are summarised in figure A.4. They consist of a central mass and a special case spring known as a 'serpentine' spring, along with the sensing comb that runs alongside the mass. In this particular case study a symmetry constraint is applied to the serpentine springs. Consequently, one spring is evolved and then mirrored in the x and y axes.

Another common approach to simulating and analysing MEMS devices is through the use of analytical lumped parameter models. Two case studies, based upon well known devices, are examined. They include the folded flexure resonator and micro gyroscope, both of which have been modelled and validated in previous work [4][53]. The analytical equations, for each device, are outlined below and have been implemented in the mathematical CAD tool MathCAD [147] and have suitably been parameterized.

Case study 4 – Folded flexure resonator

This particular resonator design is a common MEMS device, and is becoming a popular choice due to its insensitivity to buckling that may be caused by any residual stress in the polysilicon structure [3]. An electrostatic actuator or 'comb drive' is often used as an actuator as the application of a voltage across the drive results in an electrostatic force being generated. The movement or 'resonance' of the device is highly dependent on a number of factors, particularly effective mass (m_x), damping (B_x) and the stiffness of the folded flexure (K_x). These particular behavioral and physical parameters are also highly dependent on the sizing and topology of the device. The modelling of this particular device has been undertaken analytically in [3] and later in [53]. A brief overview of the main components of the model is outlined below. The parameterized model and design variables of this particular MEMS device are also shown in figure A.5.

$$K_x = \frac{2Et(W_b)^3}{{L_b}^3} \frac{{L_t}^3 + 14\alpha L_t L_b + 36\alpha^2 {L_b}^2}{4{L_t}^2 + 41\alpha L_t L_b + 36\alpha^2 {L_b}^2} \tag{A.1}$$

$$\alpha = \left(\frac{W_t}{W_b}\right)^3 \tag{A.2}$$

$$B_x = \mu\left[\left(A_s + 0.5A_t + 0.5A_b\left(\frac{1}{d} + \frac{1}{\delta}\right) + \frac{A_c}{g}\right)\right] \quad \text{(A.3)}$$

$$m_x = m_s + \frac{1}{4}m_t + \frac{12}{35}m_b \quad \text{(A.4)}$$

$$where\ m_s = \rho A_s t \qquad m_t = \rho A_t t \qquad m_b = \rho A_b t \quad \text{(A.5)}$$

These behavioural parameters of the 2D layout device are outlined by equations A.1, A.3 and A.4 for the stiffness along the x axis, damping and mass of the folded flexure resonator, respectively. The parameters, *E*, *t*, *p* corresponds to the Young's modulus of air, device thickness and density of polysilicon, while A_s, A_t, and A_b represent the total area for the device shuttle, truss and beam structures respectively. The design parameters for *d*, *g*, δ, and N correspond to spacer gap, comb finger gap, the penetration depth of airflow above the structure, and finally the number of comb fingers.

Case study 5 – Micro gyroscope

A single mass tuning fork micro gyroscope provides another suitable real world device that has been analytically modelled. The main constituents of the model are outlined below, where the full model can be found in [4]. In this case study the gyroscope is driven in a lateral direction by an electrostatic force generated by an ac voltage applied across the comb actuators (not shown). The driving direction is lateral (along x-axis) to obtain a large amplitude of deflection [4]. The Coriolis force (F_c) produced by an angular velocity (Ω) around the y-axis causes the gyroscope to oscillate in the z-direction and a capacitance change is consequently detected as an output signal. The sensitivity of the gyroscope to these changes is an important behavioural objective of designers looking to optimise this device. The topology and design variables of the micro gyroscope are outlined in figure A.5.

$$S = \frac{2mF_e\omega}{K_z K_x} \frac{1}{\sqrt{(1-{r_z}^2)^2 + \left(\frac{r_z}{Q_z}\right)^2}\sqrt{(1-{r_x}^2)^2 + \left(\frac{r_x}{Q_x}\right)^2}} \quad \text{(A.6)}$$

Electrostatic force:
$$F_e = 2\frac{2(N+1)\varepsilon_0 h}{g} V_{ac}V_{dc} \quad \text{(A.7)}$$

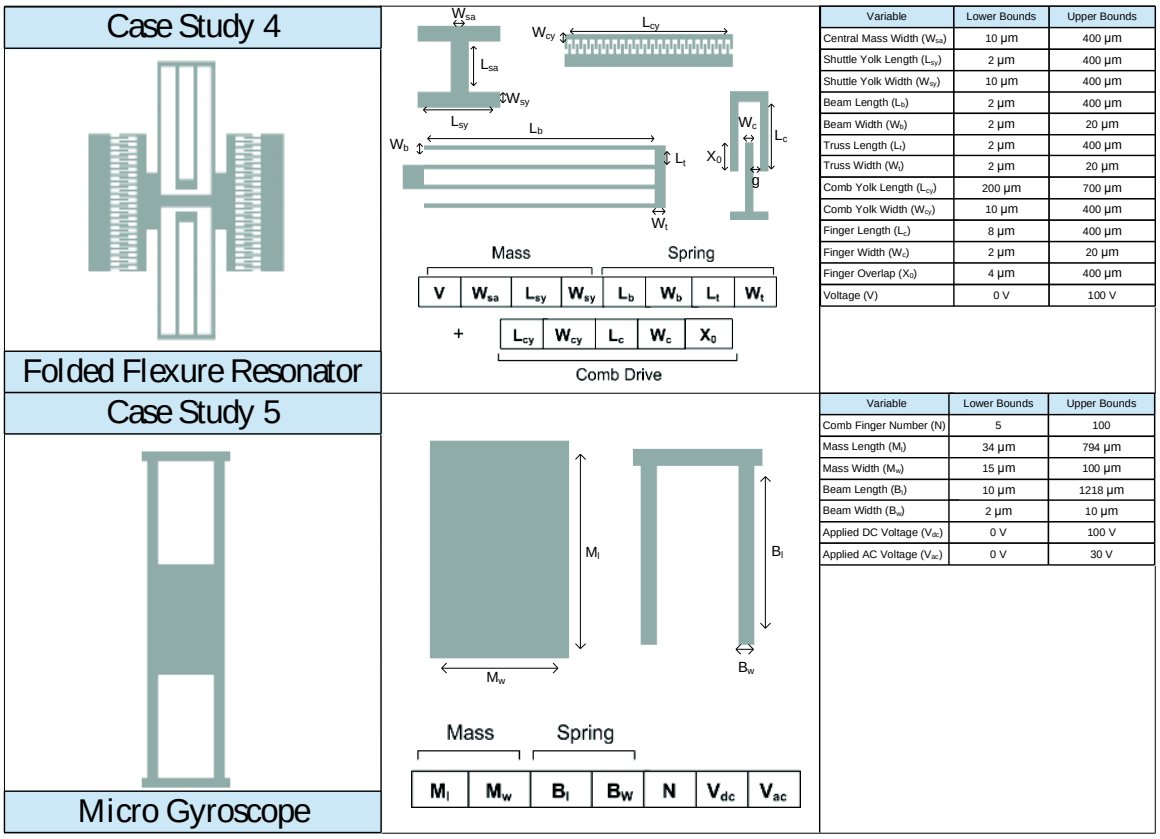
Driving mode natural frequency:
$$\omega_{rx} = \sqrt{\frac{K_x}{m}} \quad \text{(A.8)}$$

Sensing mode natural frequency:
$$\omega_{rz} = \sqrt{\frac{K_z}{m}} \quad \text{(A.9)}$$

Driving mode frequency ratio:
$$rx = \frac{\omega}{\omega_{rx}} \quad \text{(A.10)}$$

Sensing mode frequency ratio:
$$rx = \frac{\omega}{\omega_{rx}} \quad \text{(A.11)}$$

Where K_x and K_z are the stiffness in the *x* and *y* direction, *m* the mass, ε_0 is the permittivity of air, *h* finger height, *g* finger gap, ω driving frequency, Q_x and Q_z are quality factors for the driving / sensing modes outlined in [4].



Variable	Lower Bounds	Upper Bounds
Central Mass Width (W_{sa})	10 µm	400 µm
Shuttle Yolk Length (L_{sy})	2 µm	400 µm
Shuttle Yolk Width (W_{sy})	10 µm	400 µm
Beam Length (L_b)	2 µm	400 µm
Beam Width (W_b)	2 µm	20 µm
Truss Length (L_t)	2 µm	400 µm
Truss Width (W_t)	2 µm	20 µm
Comb Yolk Length (L_{cy})	200 µm	700 µm
Comb Yolk Width (W_{cy})	10 µm	400 µm
Finger Length (L_c)	8 µm	400 µm
Finger Width (W_c)	2 µm	20 µm
Finger Overlap (X_0)	4 µm	400 µm
Voltage (V)	0 V	100 V

Variable	Lower Bounds	Upper Bounds
Comb Finger Number (N)	5	100
Mass Length (M_l)	34 µm	794 µm
Mass Width (M_w)	15 µm	100 µm
Beam Length (B_l)	10 µm	1218 µm
Beam Width (B_w)	2 µm	10 µm
Applied DC Voltage (V_{dc})	0 V	100 V
Applied AC Voltage (V_{ac})	0 V	30 V

Figure A.5 Analytical modelling design optimisation problems

A.1.2 Experimental setup

Drawing on previous work undertaken in the field [4][32][42][137][53], planar MEMS devices form the basis for our evaluation of our design optimisation approach. A set of five case studies of increasing complexity have been implemented within our design optimisation environment, which forms a suitable strategy to evaluate the performance of the algorithms in question. The experiments investigate the performance of MOGAII and NSGAII for the design and optimisation of MEMS through these case studies. For each case study five experimental runs of each algorithm are conducted.

Both algorithms use some form of elitism based generational evolution and in certain case studies the chromosome representation contains both a mix of 'continuous' and 'discrete' values. The algorithms' parameters are fixed as shown in Table A.1 and correspond as much as possible to past work in the field of MEMS design optimisation and evolutionary algorithms.

Each case study contains a number of objectives, constraints and in some cases varied length representations related to the design problem at hand and each is discussed in the following sections.

Table A.1 Experimental Parameter Settings for MOGAII and NSGAII

MOGAII		NSGAII	
Probability of directional crossover	80%	Probability of polynomial crossover	80%
Probability of classical crossover	14%	Probability of mutation	1%
Probability of mutation	1%	Distribution index for crossover	20
DNA string mutation ratio	5%	Distribution index for mutation	20
Population Size	100	Population Size	100
Generations	100	Generations	100

Case study 1 – Meandering spring setup

The objectives chosen for the experiment were to evolve designs that matched a certain behaviour in this instance each spring was to have a stiffness in the x direction k_x = 2N/m, and a stiffness in the y direction K_y = 2N/m following a force applied deflection. In this instance the objectives shown in table A.2 simply become the minimization of error from the design goal of 2 N/m. The design variables are listed in figure A.5 and in this instance for both MOGAII and NSGAII contain a set of varied length chromosomes in order to accommodate the addition and removal of multiple beams within a spring design. The varied length is expressed at the phenotype stage when the decision variables are supplied to the simulation model; depending on beam number certain sections of the chromosome are clipped and ignored. However at the genotype stage the chromosome is essentially a string of decision variables and therefore requires no repair procedures s a result of genetic operators.

Table A.2 Design Objectives for Meandering Spring

Objective	Target
Stiffness K_x N/m	2.0 (Minimize Error)
Stiffness $\boldsymbol{K_y}$ N/m	2.0 (Minimize Error)

Case study 2 – Meandering resonator setup

The case study looks to evolve the 2D layout topology and sizing of a meandering resonator, built upon the meandering spring outlined previously the device consists of a central mass suspended by four springs and the design variables are outlined in figure A.5. The objectives extend those of the spring by looking to evolve designs which match a certain resonant frequency in addition to the previous stiffness objectives and are outlined in table A.3. Once again for both MOGAII and NSGAII the representation is a varied length chromosome.

Table A.3 Design Objectives for Meandering Resonator

Objective	Target
Stiffness K_x N/m	2.0 (Minimize Error)
Stiffness K_y N/m	2.0 (Minimize Error)
Frequency Rad/s	93723 Rad/s (Minimize Error)

Case study 3 – ADXL150 accelerometer setup

The ADXL150 accelerometer case study provides a real world device for application of the design optimisation framework and comparison of our two multi-objective algorithms. Both the topology and sizing of the device are included for design optimisation and the design variables are listed in figure A.5. The objectives chosen match those sought after by MEMS designers, focusing on maximizing the devices sensitivity, minimizing total area and matching a target resonant frequency as shown in table A.4.

Table A.4 Design Objectives for ADXL 150 Accelerometer

Objective	Target
Frequency Rad/s	150796 Rad/s
Total Area μm^2	Minimize
Sense Capacitance fF	Maximize

Case study 4 – Folded flexure resonator setup

A common resonator design the folded flexure resonator is used in many applications and as with the simpler meandering resonator it is important to be able to design a resonator to match certain behavioural or physical properties such as resonance or mass. Therefore three objectives have been chosen as outlined in previous work [3][53] and are shown in table A.5. These objectives for mass, stiffness and damping are useful in designing a resonator to match a certain frequency response, which is important in micro filter design [50][53]. Included in the design optimisation process are six linear and non-linear constraints tied to the layout or function of the device and are described in equations 3.2 to 3.6.

$$0 \leq L_{cy} + 2g + 2W_c \leq 700\mu m \tag{A.12}$$

$$0 \leq L_{cy} + 2L_b + 2W_t \leq 700\mu m \tag{A.13}$$

$$0 \leq 3L_t + W_{sy} + 4L_C - 2X_0 + 2W_{cy} + 2W_{ca} \leq 700\mu m \tag{A.14}$$

$$(2N+1)W_c + 2N_g \leq L_{cy} \tag{A.15}$$

$$4 \leq L_c - \left(X_0 + X_{disp}\right) \leq 200\mu m \tag{A.16}$$

Where:

$$X_{disp} = \frac{QF_{e,x}}{Kx} \tag{A.17}$$

$$F_{e,x} = 1.12\varepsilon_0 N \frac{1}{g} V^2 \tag{A.18}$$

Table A.5 Design Objectives for Folded Flexure Resonator

Objective	Target
Mass kg	5.12E-10 (Minimize Error)
Stiffness K_x N/m	2.45 (Minimize Error)
Damping	2.72E-08 (Minimize Error)

Case study 5 – Micro gyroscope setup

The micro gyroscope from the outset is a simple geometric device however the model outlined in figure A.5 does provide a suitable case study on the application of analytical models described in the literature [4] towards design optimisation. Two objectives outlined in table A.6 are applied with the goal of maximizing sensitivity of the device while minimizing its total area. The model also contains a number of design constraints, looking to evolve designs which have a quality factor greater than 5 in both the driving and sensing modes as shown in equations A.19 and A.20. The avoidance of side instability, small sensing and driving mode displacement constraints are shown in equations A.21-A.23 and there calculation can be found in detail here [4].

$$Q_x \geq 5 \tag{A.19}$$

$$Q_z \geq 5 \tag{A.20}$$

$$V_{dc} + V_{ac} < V_{si} \tag{A.21}$$

$$Z_{max} \leq 0.1\mu m \tag{A.22}$$

$$X_{max} \leq 5\mu m \tag{A.23}$$

Where:

$$K_y = \frac{4EhB_w}{B_l} \tag{A.24}$$

$$V_{si} = \frac{g^2 K_X}{2N\varepsilon_0 W_{finger}}\left(\sqrt{2\frac{K_y}{K_x} + \frac{{X_0}^2}{g^2}} - \frac{X_0}{g}\right) \tag{A.25}$$

Table A.6 Design Objectives for Micro Gyroscope

Objective	Target
Total Area μm^2	Minimize
Sensitivity $\boldsymbol{m/°/s}$	Maximize

A.1.3 Numerical results

Included in the appendix in sections A.3 and A.4 is a number of detailed tabulations of Pareto experimental results for each of the case studies undertaken. For each case study the results are listed in two sets of tables. The first is concerned with the overall efficiency of each algorithm over the five experimental runs and is divided into four sets of values. Listed are the number of Pareto solutions that were present at the end of each experimental (exp) run; the number of Pareto solutions from a particular experiment that remained when all five sets were combined; the number of Pareto solutions that remained when constraints on objective values were added; and the number of Pareto solutions from these sets that remain for each algorithm when MOGAII and NSGAII Pareto individuals are combined.

The second set of data focuses on a small set of best results obtained by each algorithm for four out of five case studies. Given the multi-objective nature of each design optimisation problem,

the best results are ranked according to a single objective only. The final set of designs used to form this data set is taken from the collated and constrained MOGAII / NSGAII Pareto results.

A set of hypervolume metric results for each case are shown in tables A.8 for the nodal and A.9 for the analytical case studies and their respective nadir points are shown below. Two sets of values are computed, the first focuses on each individual experimental run and therefore contains the best S^U, worst S^L and the mean S^M values for each algorithm. The second set of values S^C uses the final collated and constrained Pareto sets for each algorithm taken from section A.3. In all results the shaded values indicate superior values. The reference points used to compute the values are listed under each table.

In addition to the quantitative results are a number of qualitative measures of performance with the standard Pareto sets for each case study shown in figure A.6, and derived from the collated and constrained Pareto solutions found at the end of each run and listed in the tabulated Pareto experimental results of section A.3. The best results for four of the case studies are also visualised and shown in figure A.7 for the meandering spring (NSGAII), meandering resonator (MOGAII), folded flexure resonator (MOGAII), and micro gyroscope (MOGAII). The best results found for both NSGAII and MOGAII for the ADXL150 accelerometer ranked by total area are shown in figure A.8 alongside previous work in the field performed by the researchers at Berkeley University [42]. Linked to this are the design and behavioural parameters for each of these designs shown in table A.7.

The application of both the NSGAII and MOGAII heuristics to all five case studies have proved to be successful in evolving a set of optimal solutions for each of the design problems. The simplest and earliest examples within the literature for nodal optimisation the meandering spring and resonator were all able to be evolved into solutions which had a synthesis error of less than 1% for the K_x and K_y objectives respectively and in the case of frequency match the design target within 1% and this is in accordance with pervious work in the field [32][137].

The ADXL150 accelerometer case study is an advanced MEMS design optimisation problem built upon a real world device. Throughout all five experimental runs from both algorithms there is a consistent number of Pareto optimal solutions found which when constrained to have an error of less than 1% for the frequency objective and a sensitivity of greater than 133 fF still remains high.

The best results for both the NSGAII and MOGAII heuristics in figure A.8 and their variable and behavioural values in table A.7 show each algorithm to have evolved along different lines with MOGAII producing a device with a much lower total area and a sensitivity hovering above the constrained limit while NSGAII solutions are much bigger with larger sensitivity as a result.

The analytical case studies pose a different challenge however both algorithms are able to produce a large number of Pareto solutions and in the case of the folded flexure resonator NSGAII finds significantly more. However constraining the Pareto set solutions to having a synthesis error of less than 1% sees a reversal of this with MOGAII containing far more solutions which match these criteria as seen in figure A.6. Looking at the top ten results found by both algorithms in table A.17 shows far superior solutions in all three objectives compared to those found by the NSGAII algorithm.

The final micro gyroscope is similar to that of the folded flexure resonator case study with both algorithms able to find a large number of Pareto optimal solutions that decrease when constraints of solutions having sensitivity above $4e^{-11}$ are applied. MOGAII is once again superior to NSGAII when comparing the final Pareto sets as shown in figure A.6; however this does consist of the collated results from all experimental runs.

The hyper volume metric results in both tables A.8 and A.9 provide some comparative insight into the overall performance of both algorithms over each case study. Overall the performance of MOGAII is superior to that of NSGAII with the mean hyper volume performance better on four out of five case studies. However on a number of occasions, particularly for the meandering spring, resonator and micro gyroscope results NSGAII was able to produce the best set of results of a single run when compared with MOGAII, though in contrast it also produced the worst set of results in four out of five studies. The micro gyroscope is interesting as though NSGAII was able to perform better in terms of its mean hyper volume values, when comparing the set of constrained solutions collated in figure A.7 MOGAII appears to have a superior Pareto front. Where individually each MOGAII experiment was unable to prove superior to those found by NSGAII, when combined each experimental set contributes in some section towards this dominance. The ability to find solutions which lied within a very small margin of error from the design targets for each device has been shown for both algorithms however over the course of the five case studies MOGAII was once again superior to NSGAII in producing a better set of solutions when assessed using the hyper volume metric over four out of the five studies.

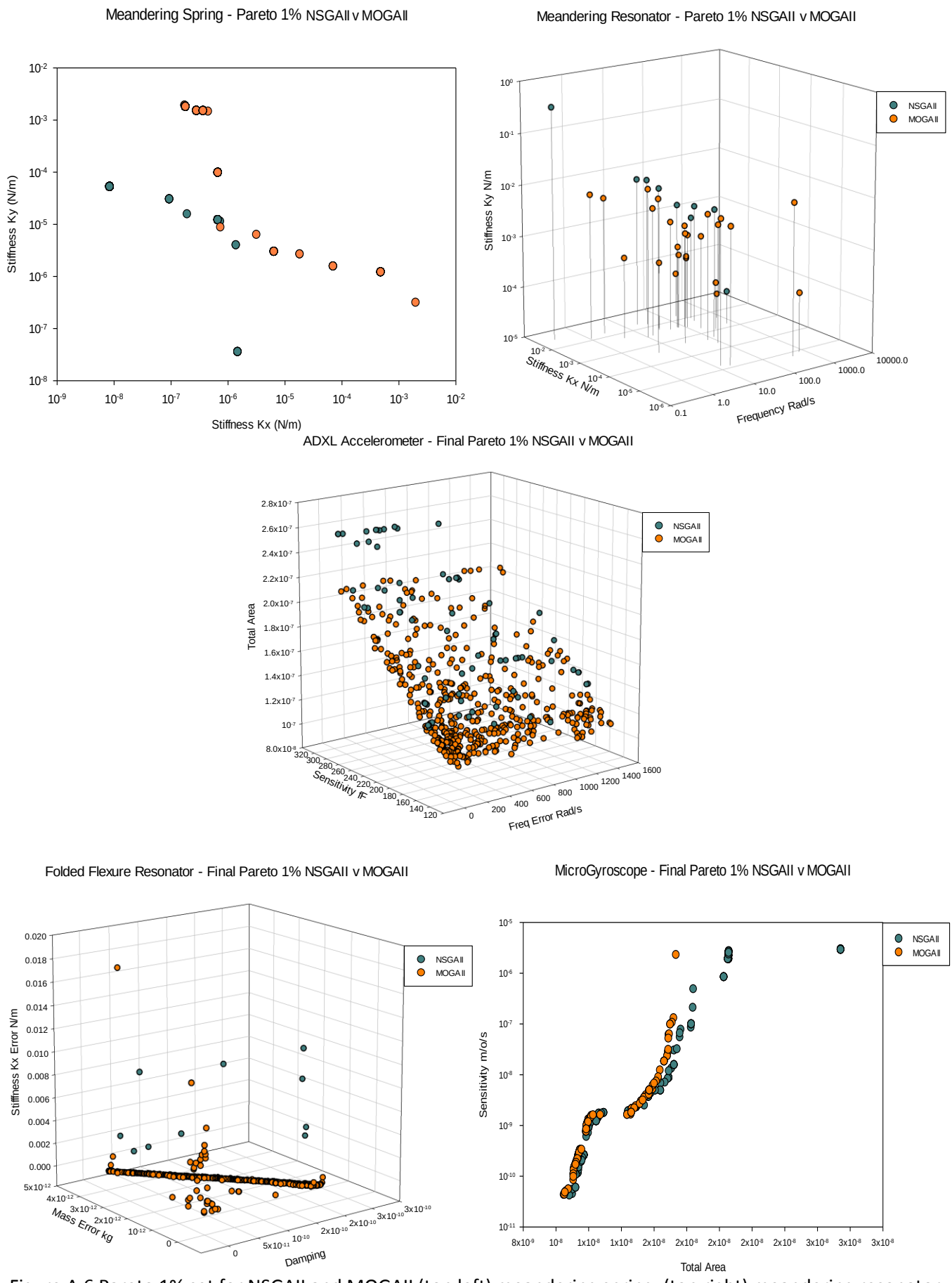


Figure A.6 Pareto 1% set for NSGAII and MOGAII (top left) meandering spring, (top right) meandering resonator, (middle) ADXL150 accelerometer, (bottom left) folded flexure resonator (bottom right) micro gyroscope

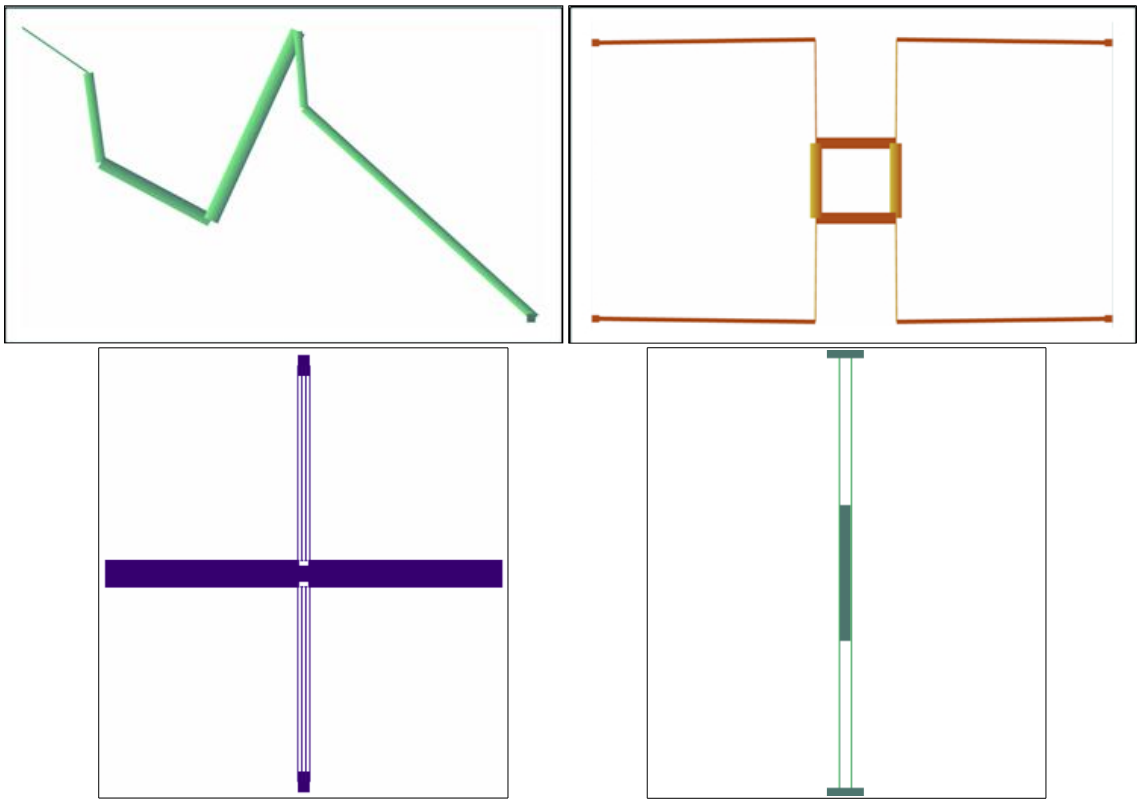


Figure A.7 Best results for (top left) meandering spring, (top right), meandering resonator, (bottom left) folded flexure resonator, (bottom right) micro gyroscope

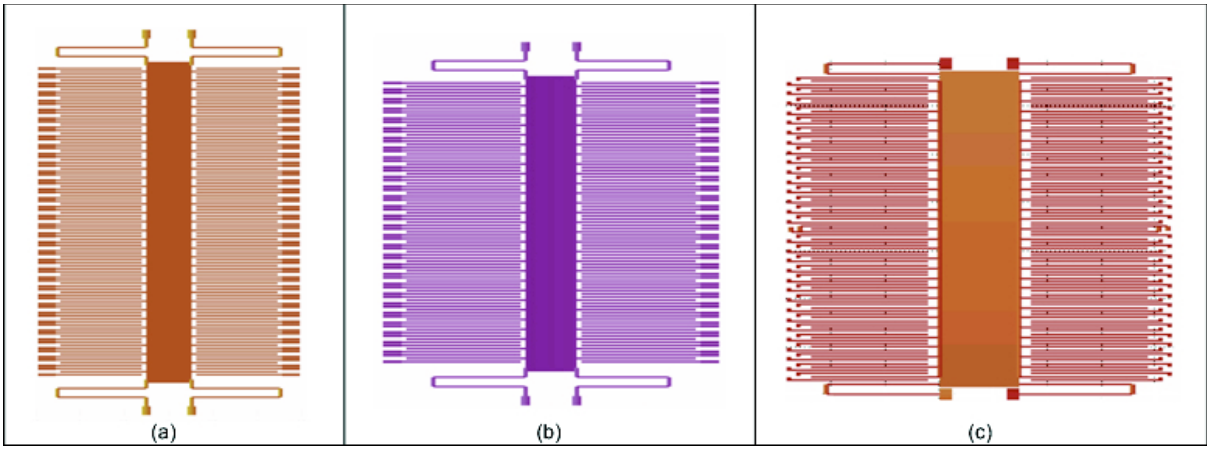


Figure A.8 Best results ADXL150 accelerometer (a) NSGAII (b) MOGAII (c) Berkeley group MOGA

Table A.7 Design and Behavioural Parameters for Best ADXL150 Accelerometer Results

Variable / Objective Values	MOGAII	NSGAII	Berkeley
Total Area	107,240 μm^2	119,660 μm^2	119,469 μm^2
Frequency	151,955 Rad/s	151,764 Rad/s	156,677 Rad/s
Sensitivity	135 fF	159 fF	152 fF
Mass length	360 μm	315 μm	329 μm
Mass width	51 μm	55 μm	75 μm
Finger length	97 μm	127 μm	118 μm
Long beam length	100 μm	99 μm	N/A
Finger cell number	35	29	31
Stiffness ratio K_x/K_y	34 N/m	33 N/m	36 N/m
Functional evaluations	10,100	10,100	50,000

Table A.8 MOGAII v NSGAII Hypervolume Metric Values for Nodal Case Studies

	Meandering Spring[1]		Meandering Resonator[2]		ADXL150 Accelerometer[3]	
	MOGAII	NSGAII	MOGAII	NSGAII	MOGAII	NSGAII
S^U	13.99999123	13.99999928	921144083268	921145055641	479.9639531	479.7192086
S^M	13.99636662	13.99528722	921110229463	920043097485	479.6307112	479.2149654
S^L	13.98630736	13.98814879	921009722766	915647911597	479.2332285	478.6435099
S^C	3.99622846E-06	3.99988905E-06	8.65610870012	8.06583378905	0.225418713	0.221176146

$*(S^U S^M S^L)^1$ [20, 0.7] $S^{C\,1}$ [0.002, 0.002] $(S^U S^M S^L)^2$ [57667, 5391, 2963] $S^{C\,2}$ [776.0, 0.032, 0.35] $(S^U S^M S^L)^3$ [$184e^{-4}$, -15, $1e^{-6}$] $S^{C\,3}$ [1508, -134, $1e^{-6}$]

Table A.9 MOGAII v NSGAII Hypervolume Metric Values for Analytical Case Studies

	Folded Flexure Resonator[1]		Micro Gyroscope[2]	
	MOGAII	NSGAII	MOGAII	NSGAII
S^U	7.080340808E-12	7.080303866e-12	1.879549580E-13	2.253480637E-13
S^M	7.080272533E-12	7.078513050E-12	4.782962378E-14	1.226925477E-13
S^L	7.080166339E-12	7.074520024e-12	1.406964178E-16	1.138720319E-16
S^C	2.266042642E-23	1.048710041e-23	2.314112081E-04	2.009851325E-14

$*(S^U S^M S^L)^1$ [$8.82e^{-8}$, $4.27e^{-10}$, $1.88e^{5}$] $S^{C\,1}$ [$2.71e^{-8}$, $4.65e^{-12}$, $1.8e^{-2}$] $(S^U S^M S^L)^2$ [$-1e^{-13}$, $1e^{-7}$] $S^{C\,2}$ [$-4e^{-11}$, $2.75e^{-8}$]

A.2 Discussion and conclusions

Traditional design and optimisation of MEMS can be slow, costly and often unable to actually lead to the most optimal solution. Over the recent decades the increase in the number of modelling and simulation tools has helped to automate the process of design, giving designers tools to build 'in-silica' devices which no longer have to be fabricated to be tested. However hand driven optimisation or local gradient based search algorithms are still common practice among designers. With the ever increasing complexity of MEMS design this approach will struggle. Recently there has been a shift into the use of more stochastic algorithms and given the nature of most engineering problems these are at a multi-objective level. The coupling of both automated modelling and analysis with more powerful stochastic multi-objective algorithms could substantially help to automate the design optimisation of MEMS.

This section looked to construct a new MEMS multi-objective design optimisation framework through the successful integration of a design optimisation platform [146] and a host of computer-aided design tools. The framework has been evaluated over a number of MEMS design synthesis and optimisation case studies using state of the art MOEAs and providing successful results in all cases.

The results from the last three case studies in particular show improvement over the previous state of the art designs within the literature [4][42][53] In comparison to those designs found by the Berkeley MEMS group [42] the optimal design for the ADXL150 accelerometer provided by the platform was superior through its reduced total area and sensitivity above 133fF. This result also required a fifth of the functional evaluations than those used previously, however the Berkeley results were optimised with a larger set of weighted objectives which may have restricted performance.

The analytical folded flexure resonator has been used before as a model for design optimisation [53]. In this instance a weighted multi-objective genetic algorithm was used. The results produced here by both algorithms matched the performance outlined in [53] but at a significant reduction to 10,100 functional evaluations, from 137,500 evaluations as described in the

literature. Finally, in the case of the micro gyroscope study a fair and direct comparison is challenging as the previous work [4] focused on a simulated annealing algorithm and only contained a single objective of maximizing the sensitivity of the device. However the optimal design produced does match the performance in terms of sensitivity at a reduction in total area which can be seen as a successful application of multi-objective genetic algorithms to MEMS design synthesis and optimisation.

A.3 **Pareto experimental results**

Table A.10 MOGAII v NSGAII experimental results for the meandering spring

MOGAII				NSGAII			
Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol <1% Error per Obj	Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol <1% Error per Obj
1	4316	2299	0	1	2944	1	0
2	26	0	0	2	2322	0	0
3	910	910	910	3	2866	0	0
4	2919	2873	1	4	2886	0	0
5	1920	0	0	5	209	209	209
Total	10091	6082	911	Total	11227	210	209
Total MOGAII v NSGAII	-	-	1	Total MOGAII v NSGAII	-	-	209

Constraints placed upon the final collated Pareto sets for the meandering spring are to have an error of less than 1% for both the K_x and K_y objectives.

Table A.11 MOGAII v NSGAII experimental results for the meandering resonator

MOGAII				NSGAII			
Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol <1% Error per Obj	Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol <1% Error per Obj
1	66	22	7	1	220	10	0
2	31	18	12	2	16	0	0
3	49	1	0	3	87	82	8
4	215	1	0	4	182	31	0
5	42	20	10	5	164	1	0
Total	403	62	29	Total	669	124	8
Total MOGAII v NSGAII	-	-	29	Total MOGAII v NSGAII	-	-	2

Constraints placed upon the final collated Pareto sets for the meandering resonator are to have an error of less than 1% for the K_x and K_y and frequency objectives.

Table A.12 MOGAII v NSGAII experimental results for the ADXL150 accelerometer

MOGAII				NSGAII			
Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol > 133 fF + <1% Freq Error per	Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol > 133 fF + <1% Freq Error per
1	1525	551	47	1	1741	684	36
2	1389	646	69	2	1781	289	34
3	1613	547	88	3	1298	382	7
4	1494	940	146	4	1325	857	19
5	1464	695	134	5	1229	449	22
Total	7485	3379	484	Total	7374	2661	118
Total MOGAII v NSGAII	-	-	484	Total MOGAII v NSGAII	-	-	18

Constraints placed upon the final collated Pareto sets for the ADXL150 accelerometer are to have an error of less than 1% for the frequency objective and sensitivity above 133fF.

Table A.13 MOGAII v NSGAII experimental results for the folded flexure resonator

MOGAII				NSGAII			
Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol <1% Error per Obj	Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol <1% Error per Obj
1	145	0	0	1	1032	29	0
2	235	57	0	2	671	83	33
3	130	20	9	3	1436	0	0
4	275	1	0	4	1191	4	0
5	682	682	682	5	748	509	0
Total	1467	760	691	Total	5078	625	33
Total MOGAII v NSGAII	-	-	691	Total MOGAII v NSGAII	-	-	0

Constraints placed upon the final collated Pareto sets for the folded flexure resonator are to have an error of less than 1% for both the K_x, mass and damping objectives.

Table A.14 MOGAII v NSGAII experimental results for the micro gyroscope

MOGAII				NSGAII			
Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol > $4e^{-11}$ Sensitivity	Exp	No of Pareto Sol in Exp	No of Pareto Sol Collated	No Sol > $4e^{-11}$ Sensitivity
1	227	14	2	1	163	76	0
2	295	145	98	2	356	0	0
3	422	306	10	3	99	17	17
4	346	70	56	4	171	0	0
5	274	41	5	5	239	198	151
Total	1564	576	171	Total	1028	291	168
Total MOGAII v NSGAII	-	-	162	Total MOGAII v NSGAII	-	-	34

Constraints placed upon the final collated Pareto sets for the micro gyroscope are to have an a sensitivity above $4e^{-11}$.

A.4 **Top 10 best experimental results**

Table A.15 MOGAII v NSGAII top 10 frequency results for culled < 1% set for the meandering resonator

MOGAII				NSGAII			
Exp	Freq Error Rad/s	Kx Error N/m	Kx Error N/m	Exp	Freq Error Rad/s	Kx Error N/m	Kx Error N/m
1	1.07	1.253E-02	6.911E-03	3	20.15	1.851E-02	8.094E-03
5	2.75	1.644E-02	4.814E-03	3	30.08	1.445E-02	7.766E-03
5	3.93	5.284E-03	3.878E-04	3	33.24	6.372E-03	6.175E-03
1	4.04	3.722E-04	9.679E-03	3	72.17	4.353E-03	2.785E-03
1	4.04	3.722E-04	9.679E-03	3	122.65	4.875E-04	3.309E-03
1	4.04	3.722E-04	9.679E-03	3	245.77	5.755E-03	2.047E-03
1	4.04	3.722E-04	9.679E-03	3	279.95	8.898E-03	1.072E-03
2	10.45	3.079E-03	8.302E-03	3	668.50	1.678E-03	4.429E-03
2	13.29	1.644E-05	6.495E-03	-	-	-	-
2	13.91	3.165E-03	3.322E-03	-	-	-	-

The top ten results are ranked by the frequency error objective.

Table A.16 MOGAII v NSGAII top 10 total area results for culled < 1% set for the ADXL150 accelerometer

MOGAII				NSGAII			
Exp	Freq Error Rad/s	Sensitivity fF	Total Area m^2	Exp	Freq Error Rad/s	Sensitivity fF	Total Area m^2
4	1.1598E3	135	1.0724E-07	4	9.6521E2	150	1.1966E-07
4	1.1148E3	135	1.0735E-07	1	8.5409E2	162	1.2333E-07
5	5.0848E2	135	1.0741E-07	1	5.3404E2	164	1.2421E-07
1	4.1785E1	136	1.0779E-07	1	5.3404E2	164	1.2421E-07
1	4.1785E1	136	1.0779E-07	1	1.4482E1	173	1.3015E-07
1	4.1785E1	136	1.0779E-07	2	3.6893E2	174	1.3108E-07
5	1.1351E3	136	1.0801E-07	2	5.2036E0	173	1.3228E-07
4	1.2665E3	138	1.0865E-07	2	5.2036E0	173	1.3228E-07
5	1.2586E2	136	1.0911E-07	2	5.2036E0	173	1.3228E-07
1	1.1894E3	139	1.0914E-07	2	5.2036E0	173	1.3228E-07

The top ten results are ranked by the total area objective.

Table A.17 MOGAII v NSGAII top 10 mass error results for culled < 1% set for the folded flexure resonator

MOGAII				NSGAII			
Exp	Mass Error kg	K_x Error N/m	Damping Error	Exp	Mass Error kg	K_x Error N/m	Damping Error
5	1.1408E-16	9.9440E-04	5.1941E-12	2	4.6222E-13	1.1146E-02	1.8989E-10
5	1.1408E-16	9.9440E-04	5.1941E-12	2	4.6222E-13	1.1146E-02	1.8989E-10
5	1.3147E-16	9.9780E-04	5.1929E-12	2	4.6222E-13	1.1146E-02	1.8989E-10
5	6.6245E-16	3.6464E-04	1.8797E-10	2	4.6222E-13	1.1146E-02	1.8989E-10
5	6.6245E-16	3.6464E-04	1.8797E-10	2	4.6222E-13	1.1146E-02	1.8989E-10
5	6.6245E-16	3.6464E-04	1.8797E-10	2	1.7607E-12	7.1956E-03	2.4962E-10
5	1.2466E-15	1.7774E-04	1.7824E-11	2	1.7607E-12	7.1956E-03	2.4962E-10
5	3.1284E-15	1.7171E-06	1.9930E-10	2	1.7607E-12	7.1956E-03	2.4962E-10
5	3.1284E-15	1.7171E-06	1.9930E-10	2	1.8592E-12	9.8004E-03	1.1358E-10
5	3.1284E-15	1.7171E-06	1.9930E-10	2	2.0491E-12	2.4448E-03	2.7081E-10

The top ten results are ranked by the mass error objective.

Table A.18 MOGAII v NSGAII top 10 total area results for culled > $4e^{-11}$ sensitivity set for the micro gyroscope

MOGAII			NSGAII		
Exp	Total Area m^2	Sensitivity $m/°/s$	Exp	Total Area m^2	Sensitivity $m/°/s$
2	1.0491E-08	4.2503E-11	3	1.0831E-08	4.1384E-11
2	1.0523E-08	4.3100E-11	3	1.1022E-08	4.4459E-11
2	1.0523E-08	4.3098E-11	3	1.1301E-08	5.4452E-11
2	1.0523E-08	4.3098E-11	5	1.1221E-08	5.9067E-11
2	1.0530E-08	4.3330E-11	5	1.1221E-08	5.9067E-11
2	1.0530E-08	4.4146E-11	5	1.1236E-08	1.0565E-10
2	1.0547E-08	4.4248E-11	5	1.1254E-08	1.0820E-10
2	1.0553E-08	4.6297E-11	5	1.1360E-08	1.1369E-10
2	1.0565E-08	4.7455E-11	5	1.1367E-08	1.1453E-10
2	1.0565E-08	4.7455E-11	5	1.1391E-08	1.3014E-10

The top ten results are ranked by the total area objective.

B

Appendix

Bandpass filter case study validation

B.1 Bandpass filter experimental results

The following section holds the bandpass filter responses for each of the three case studies (656 Hz, 20 kHz, and 100 kHz), and for each population set (100, 20).

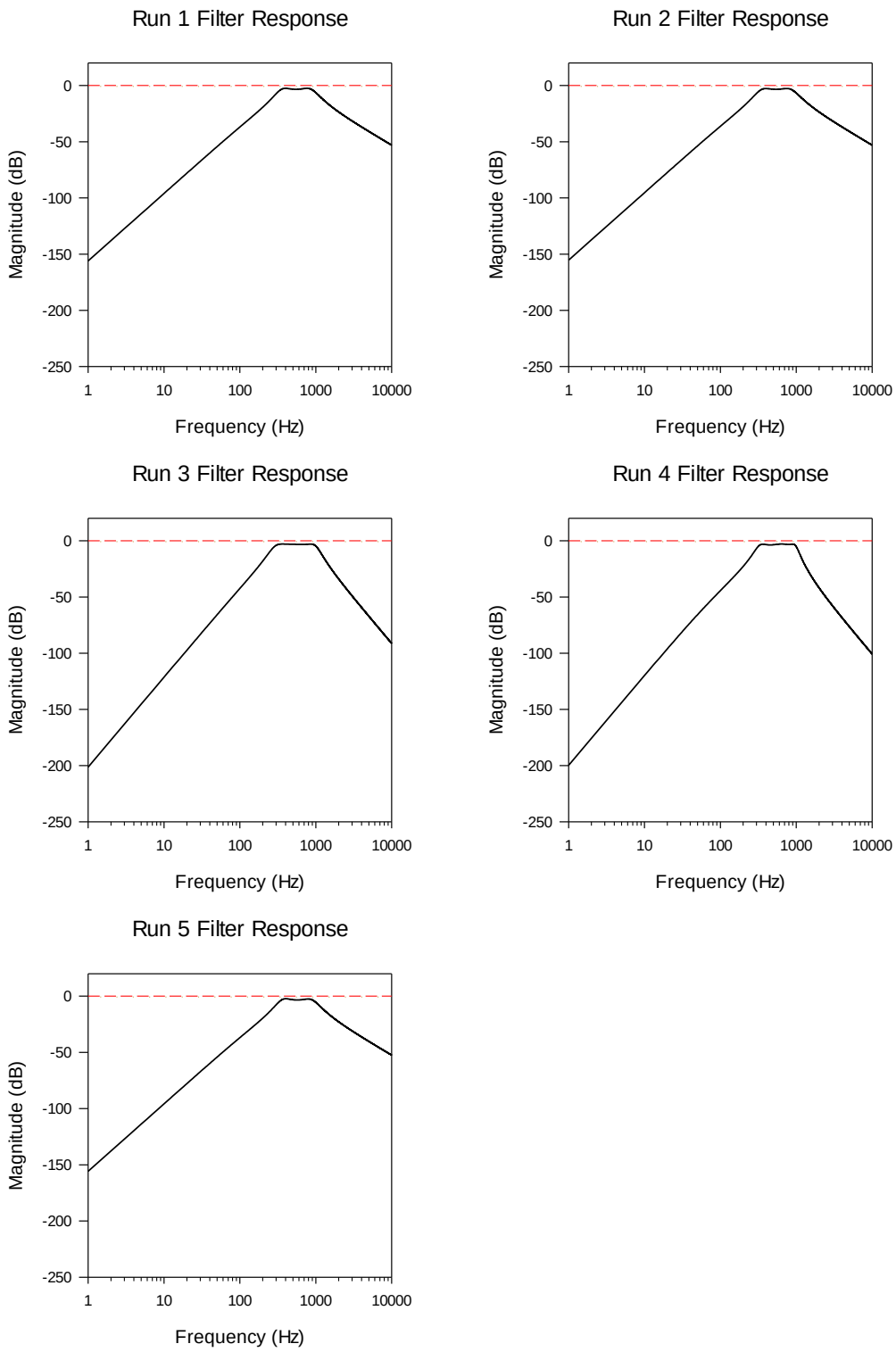


Figure B.1 Bandpass filter validation 656 Hz run 1 – 5 best filter response ranked by filter frequency objective for NSGAII 100 population set

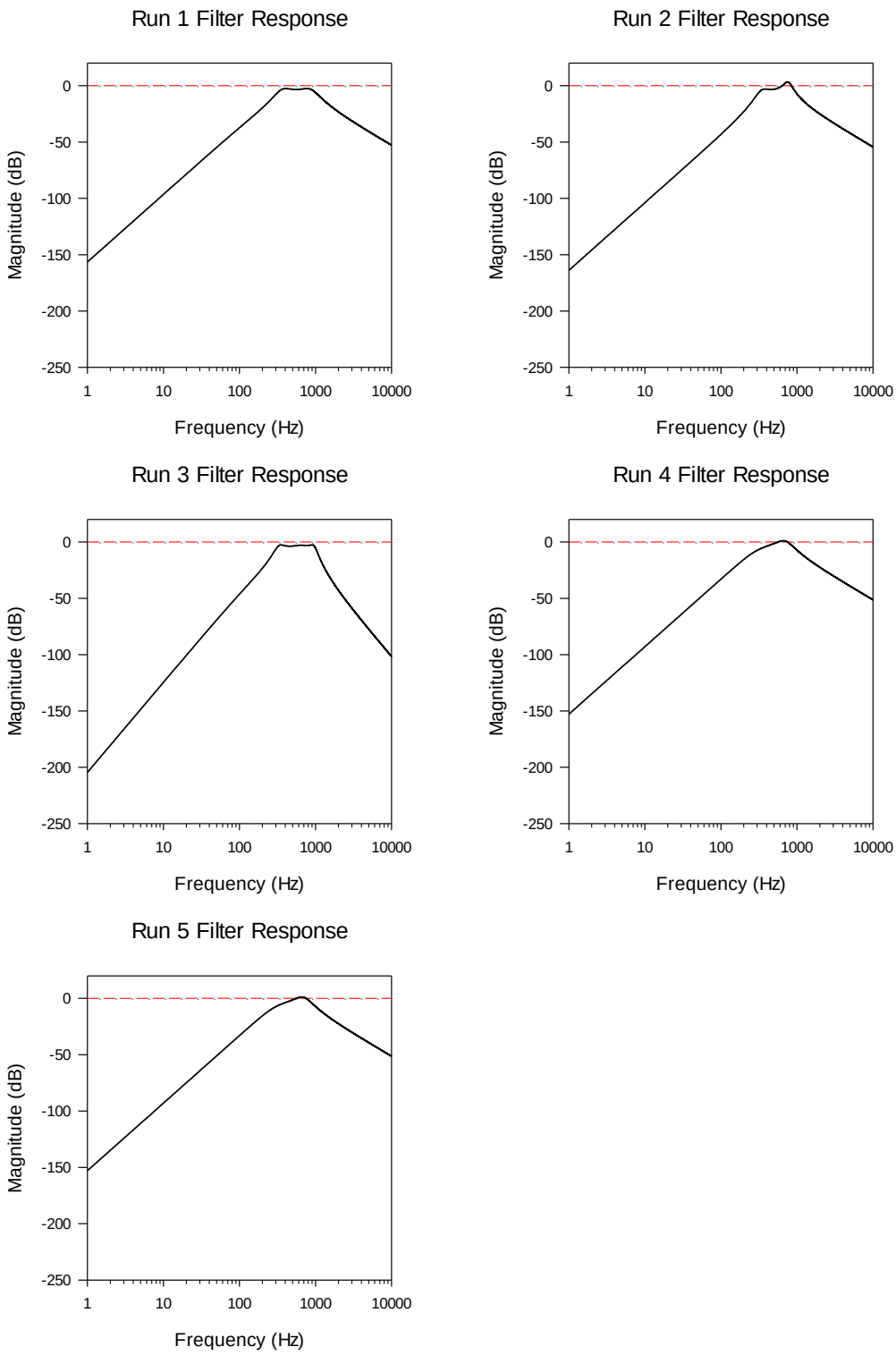


Figure B.2 Bandpass filter validation 656Hz run 1 – 5 best filter response ranked by filter frequency objective for NSGAII 20 population set

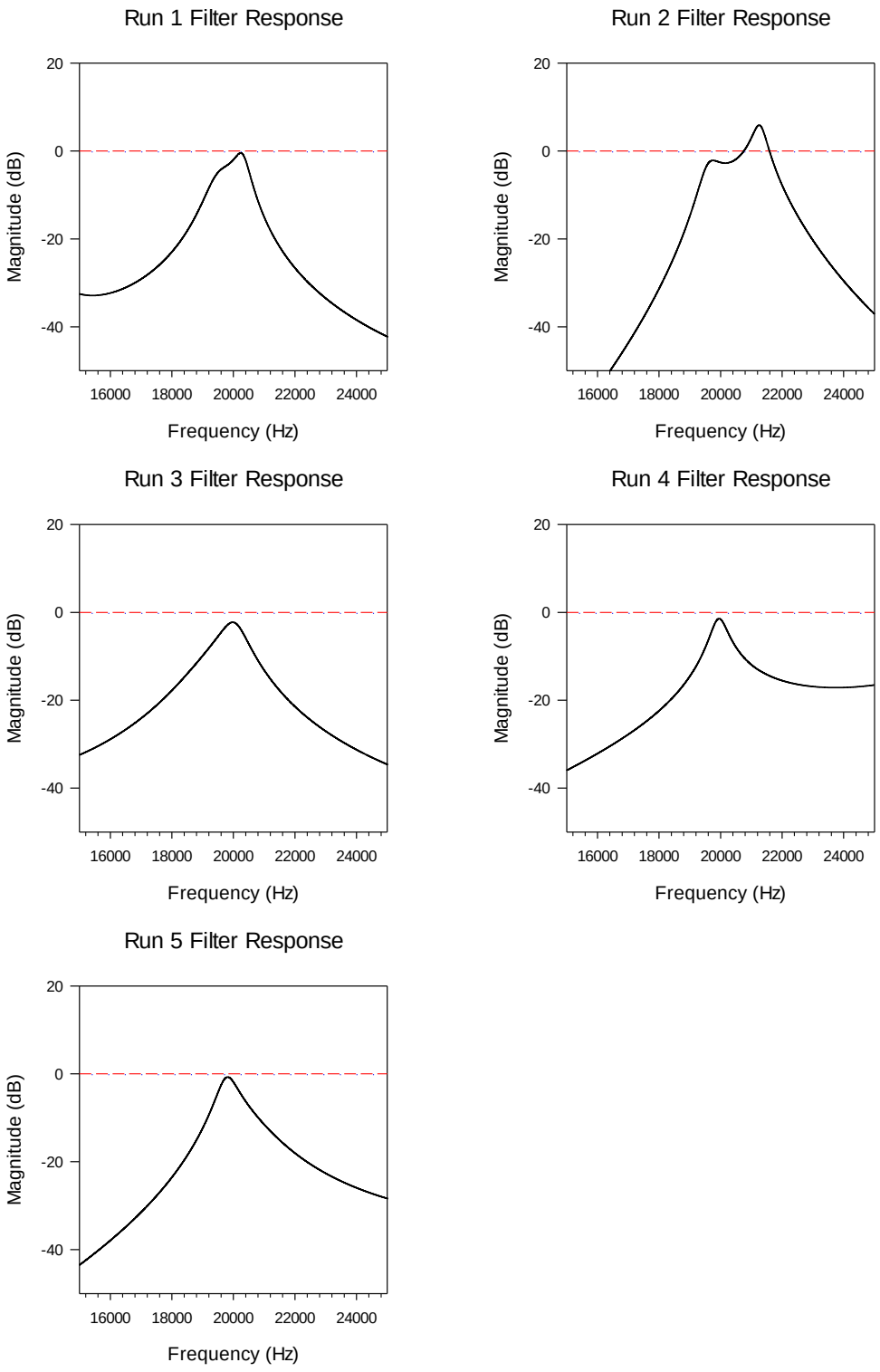


Figure B.3 Bandpass filter validation 20 kHz run 1 – 5 best filter response ranked by filter frequency objective for NSGAII 100 population set

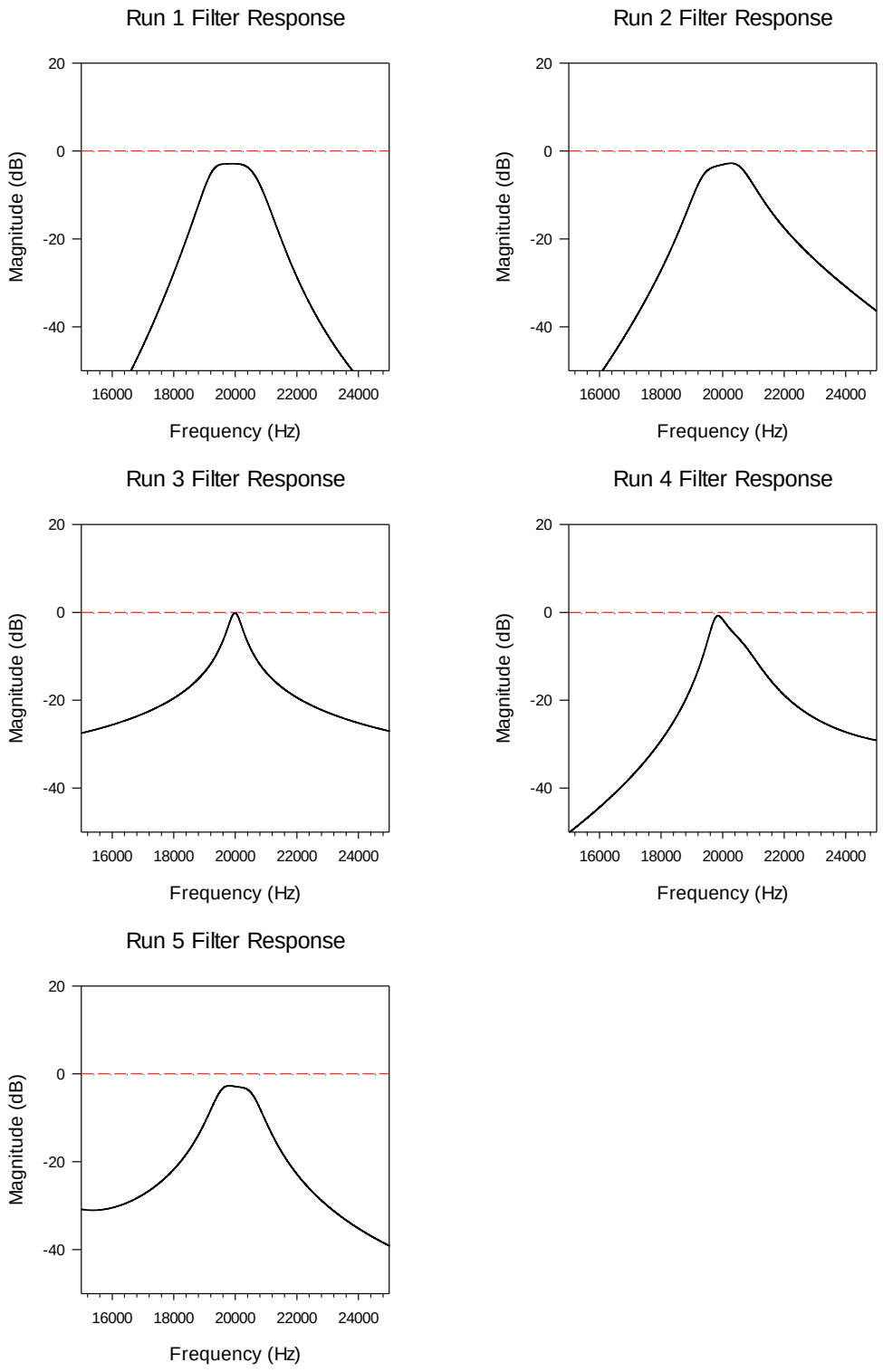


Figure B.4 Bandpass filter validation 20 kHz run 1 – 5 best filter response ranked by filter frequency objective for NSGAII 20 population set

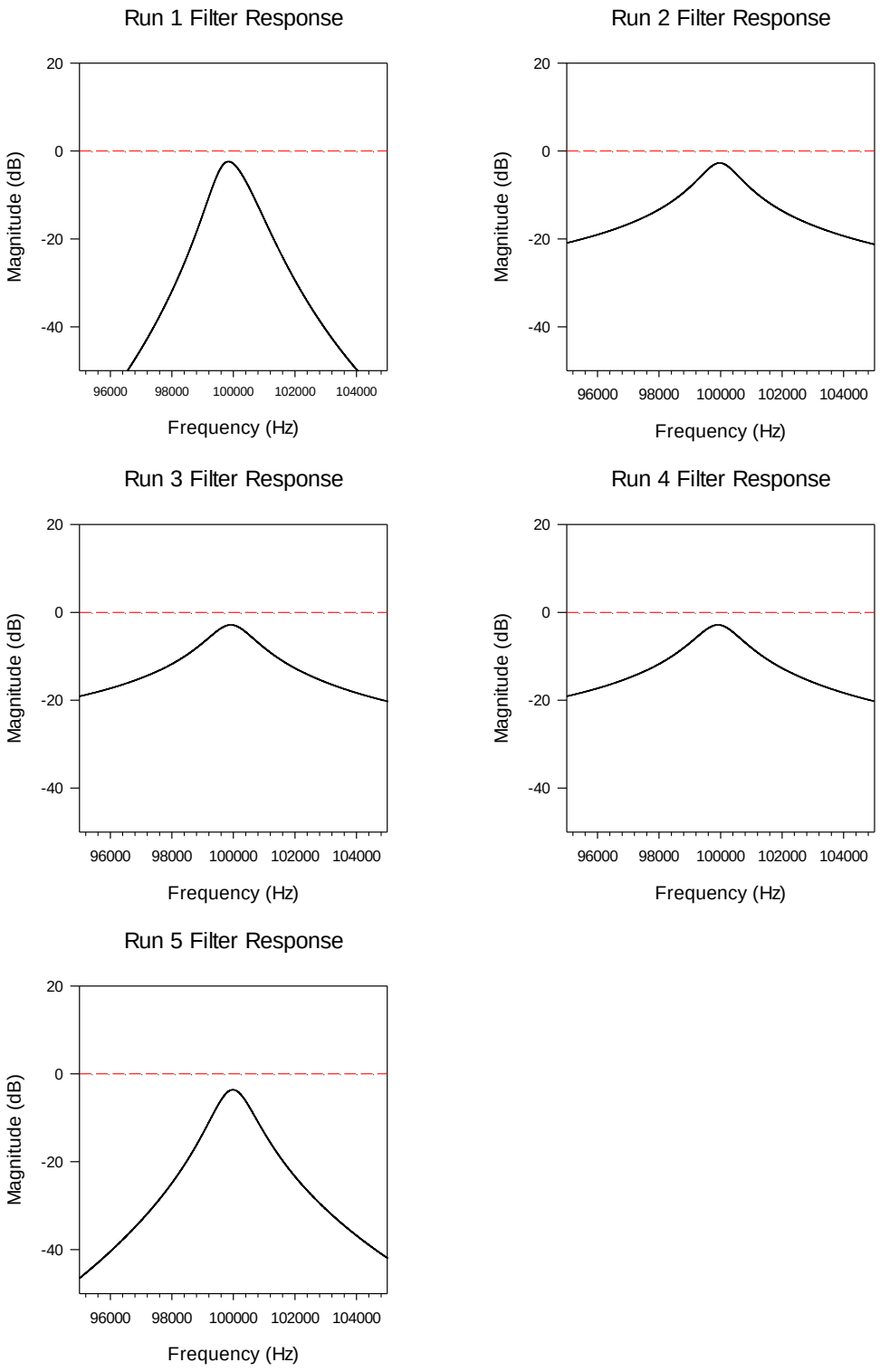


Figure B.5 Bandpass filter validation 100 kHz run 1 – 5 best filter response ranked by filter frequency objective for NSGAII 100 population set

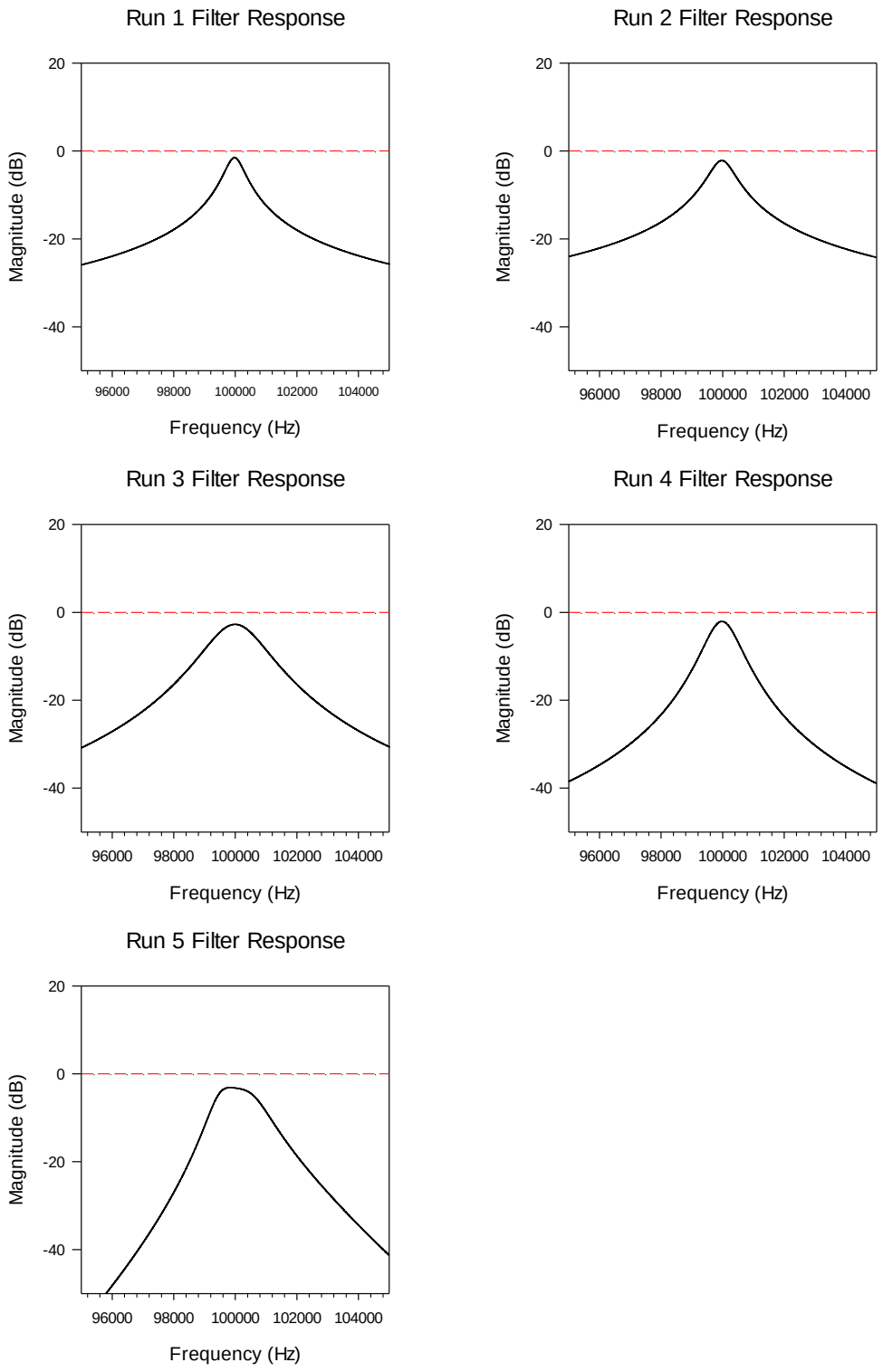


Figure B.6 Bandpass filter validation 100 kHz run 1 – 5 best filter response ranked by filter frequency objective for NSGAII 20 population set

C

Appendix

Uni-level design optimisation results

C.1 System level experimental results

The following section holds system level results for the single level, multi-level evaluation, multi-level parameterization and multidisciplinary optimisation strategies. These include final population sets and the best filter results achieved by each algorithm and strategy, along with some strategy specific analysis results.

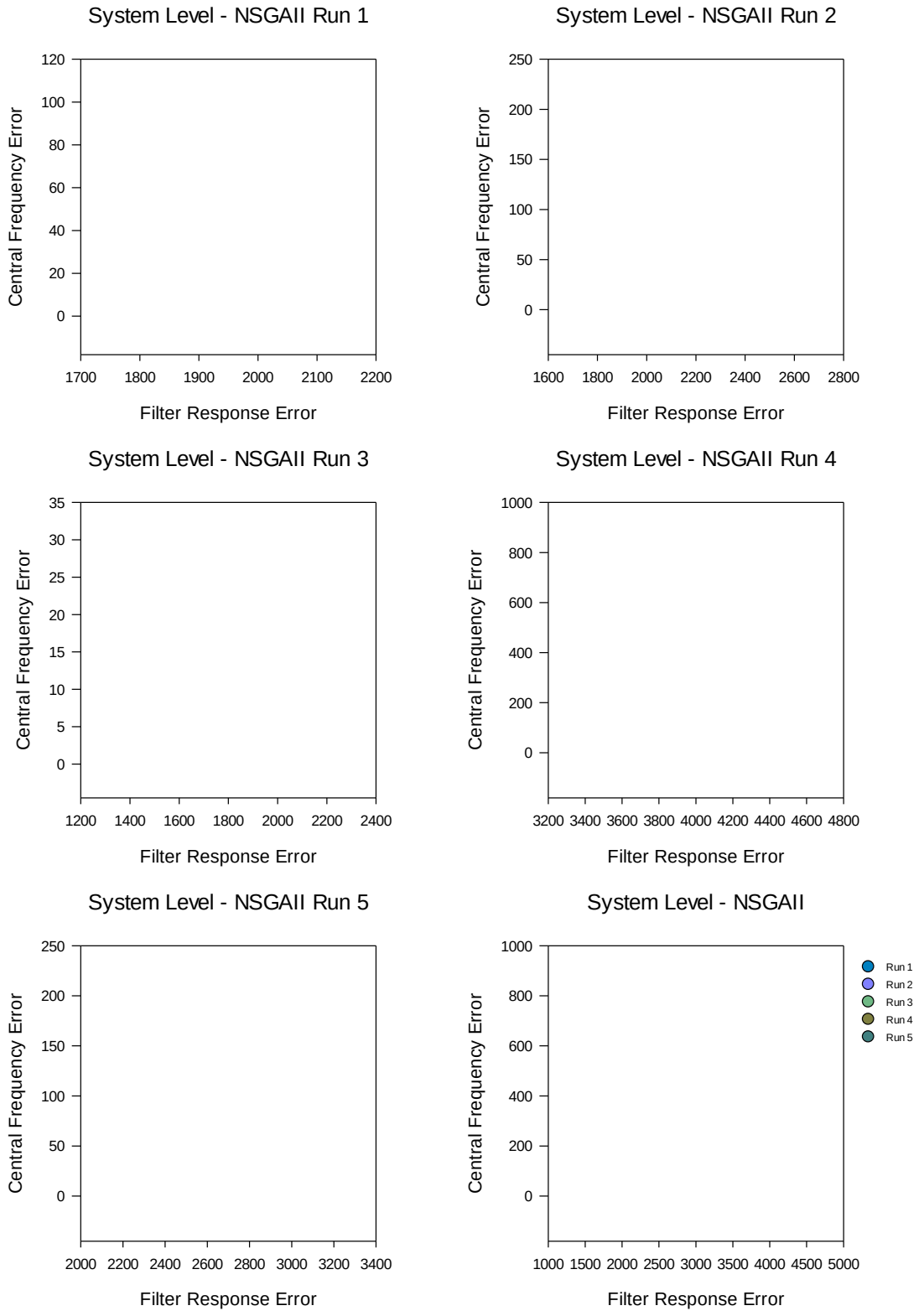


Figure C.1.1 System level run 1 – 5 final population sets for NSGAII

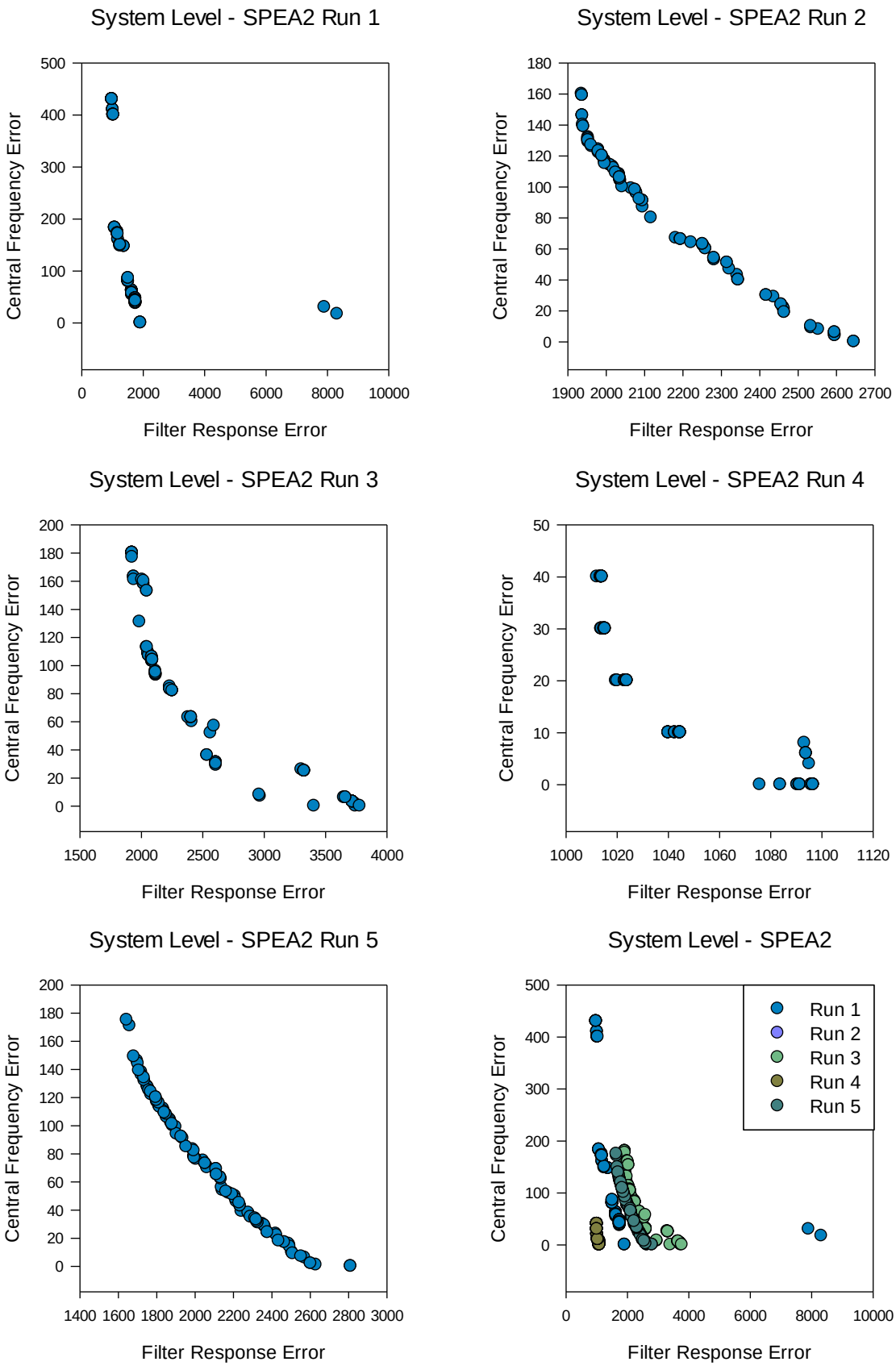


Figure C.2 System level run 1 – 5 final population sets for SPEA2

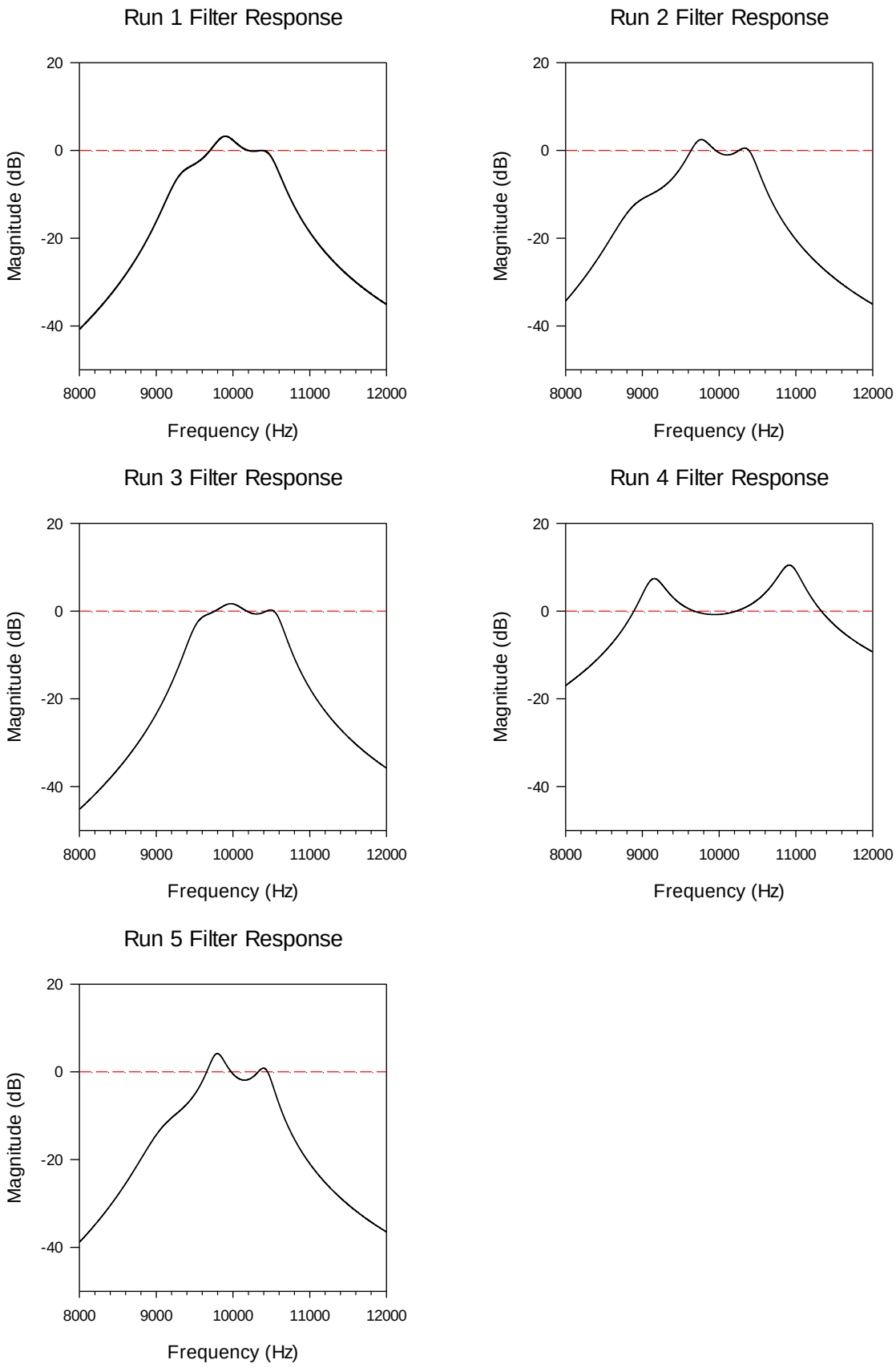


Figure C.3 System level run 1 – 5 best filter response ranked by filter frequency objective for NSGAII

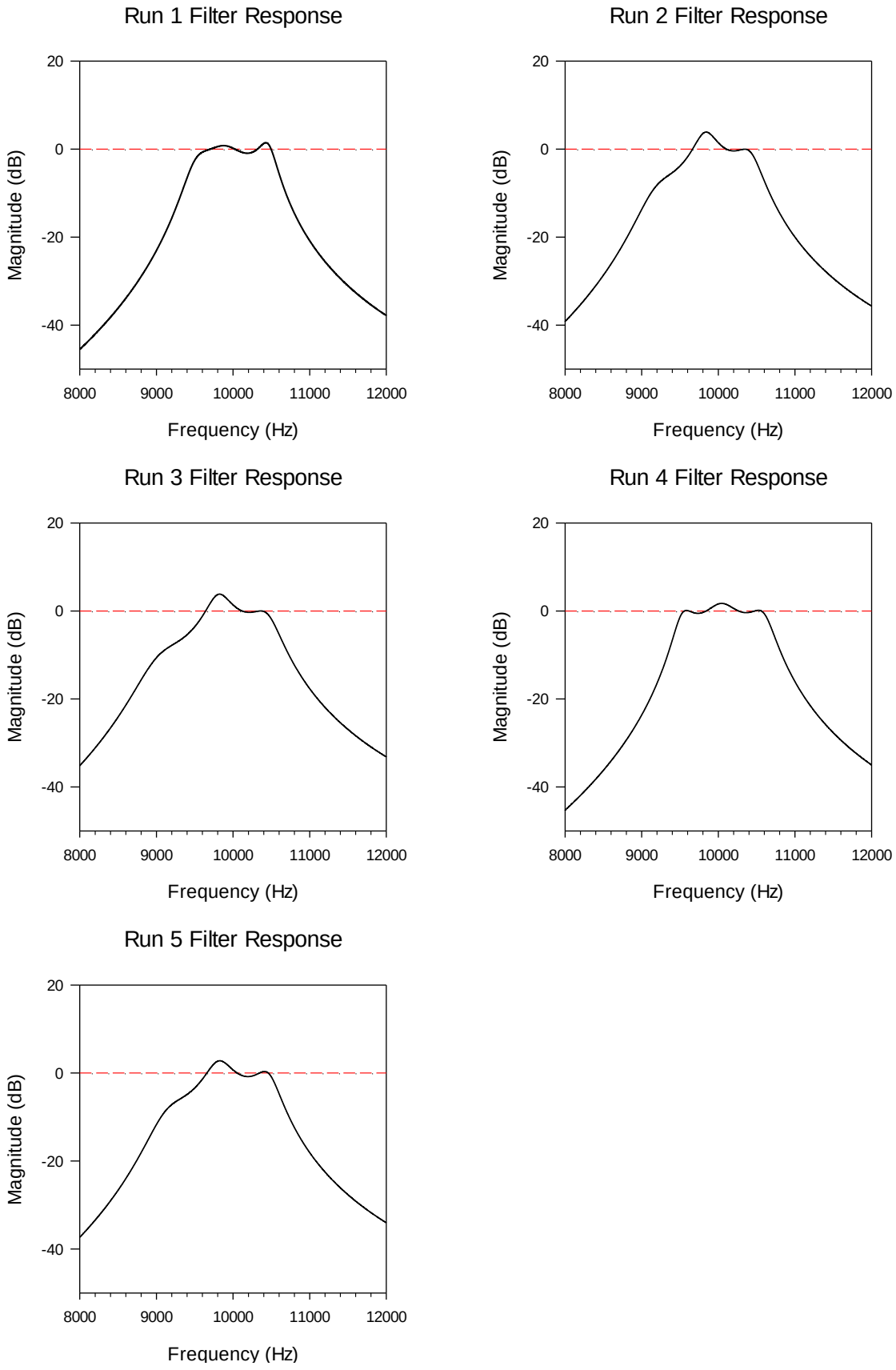


Figure C.4 System level run 1 – 5 best filter response ranked by filter frequency objective for SPEA2

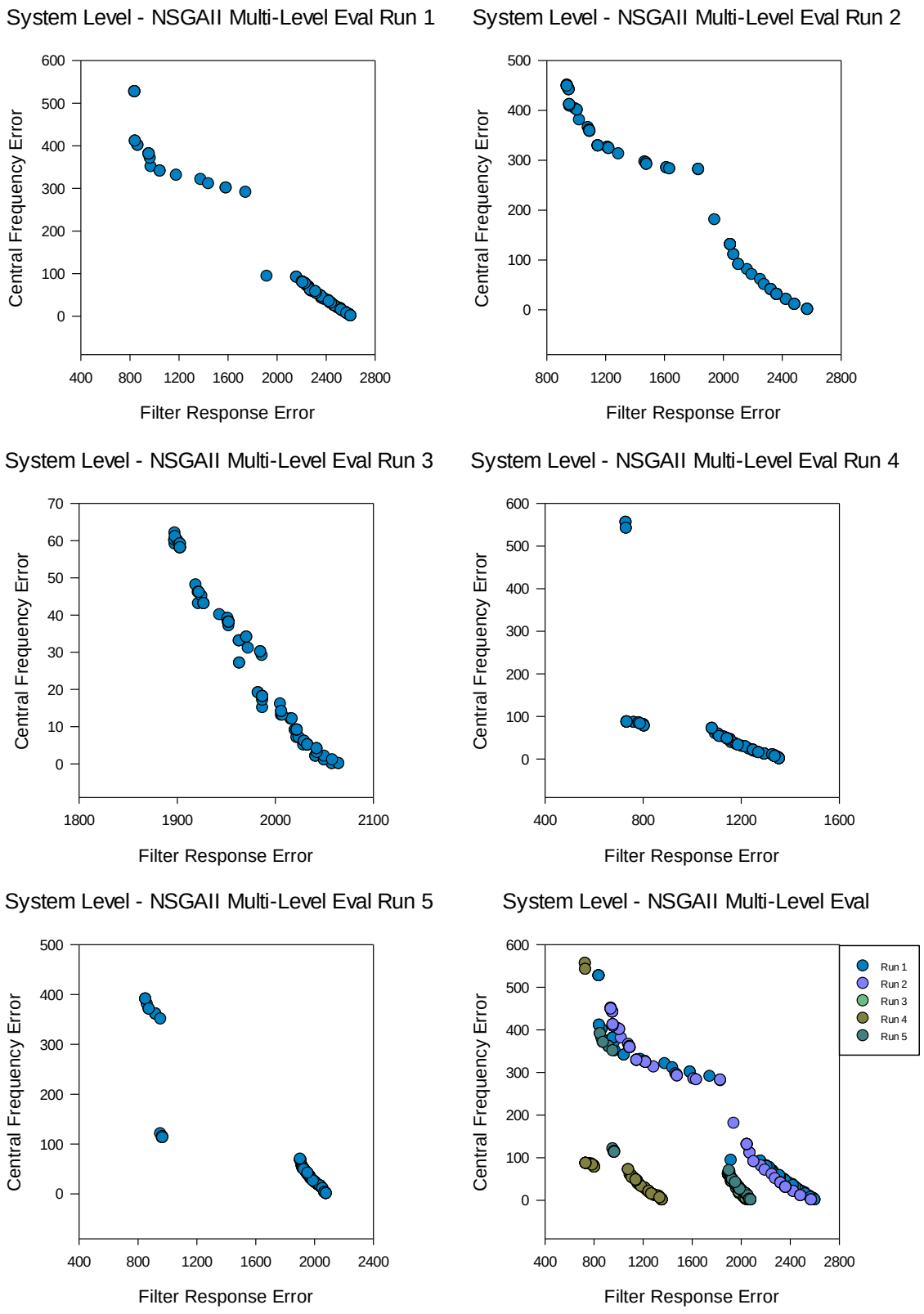


Figure C.5 System level run 1 – 5 final population sets for NSGAII multi-level evaluation

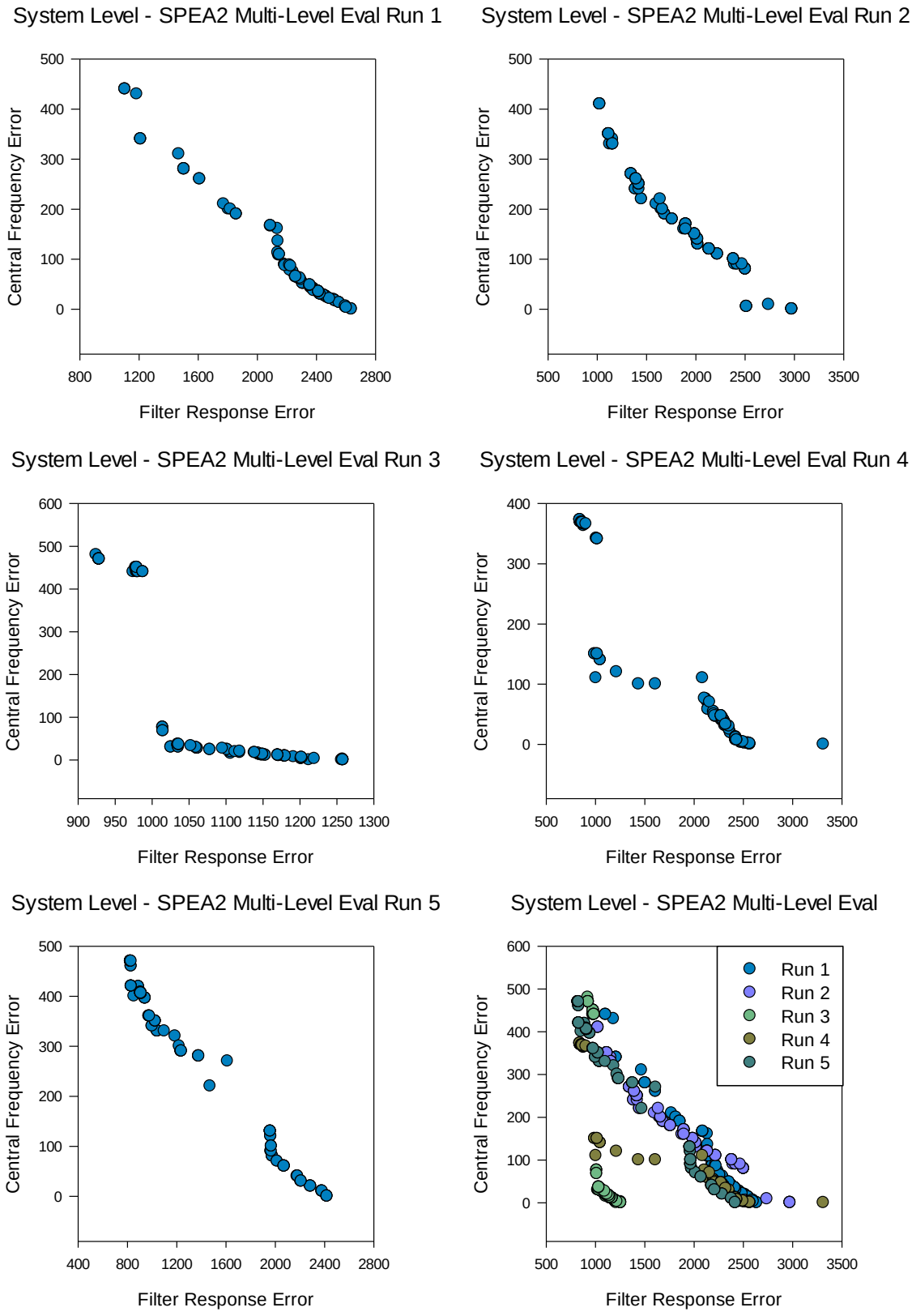


Figure C.6 System level run 1 – 5 final population sets for SPEA2 multi-level evaluation

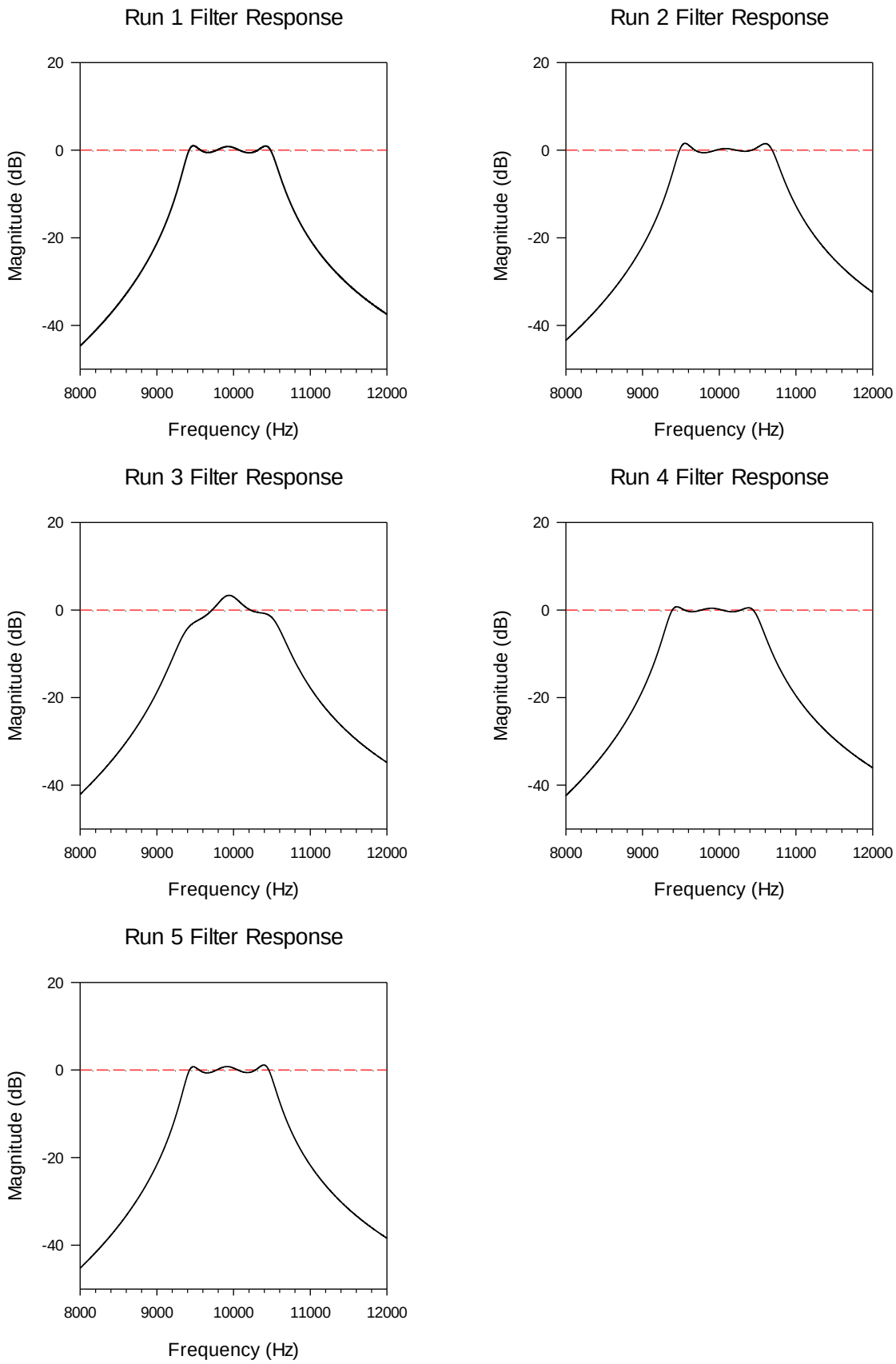


Figure C.7 System level run 1 – 5 best filter response ranked by filter frequency objective for NSGAII multi-level evaluation

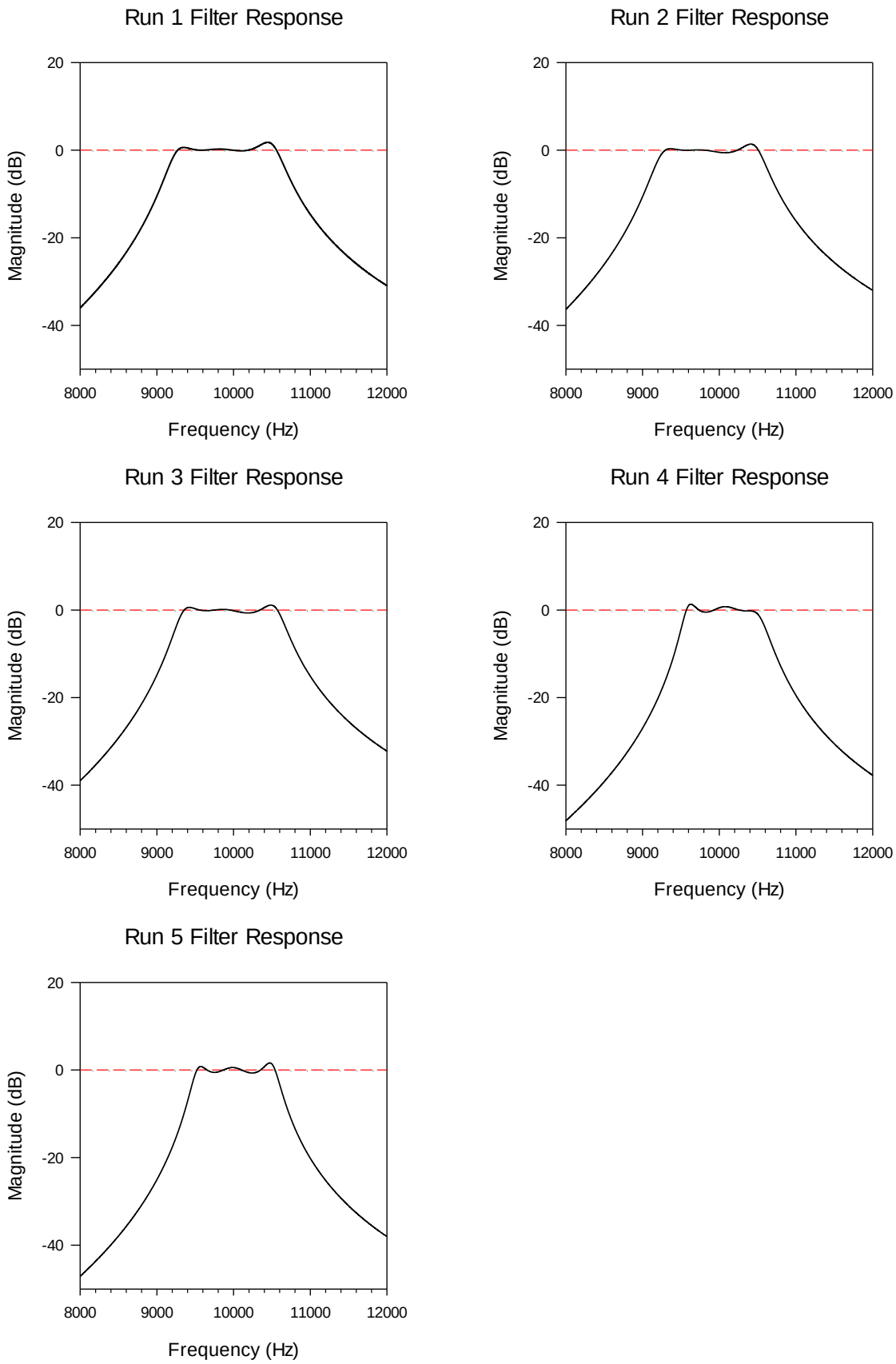


Figure C.8 System level run 1 – 5 best filter response ranked by filter frequency objective for SPEA2 multi-level evaluation

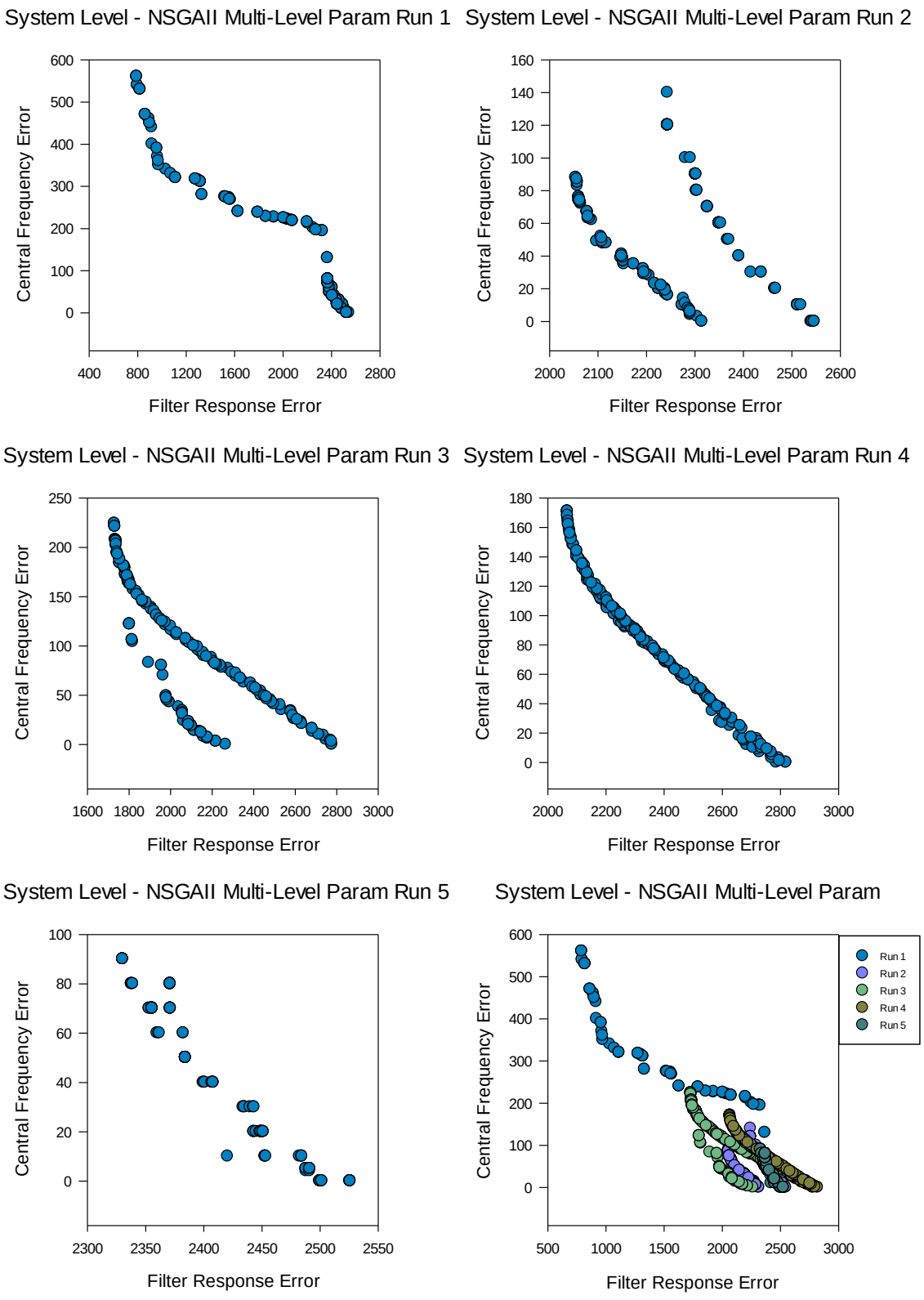


Figure C.9 System level run 1 – 5 final population sets (high + low) for NSGAII multi-level parameterization

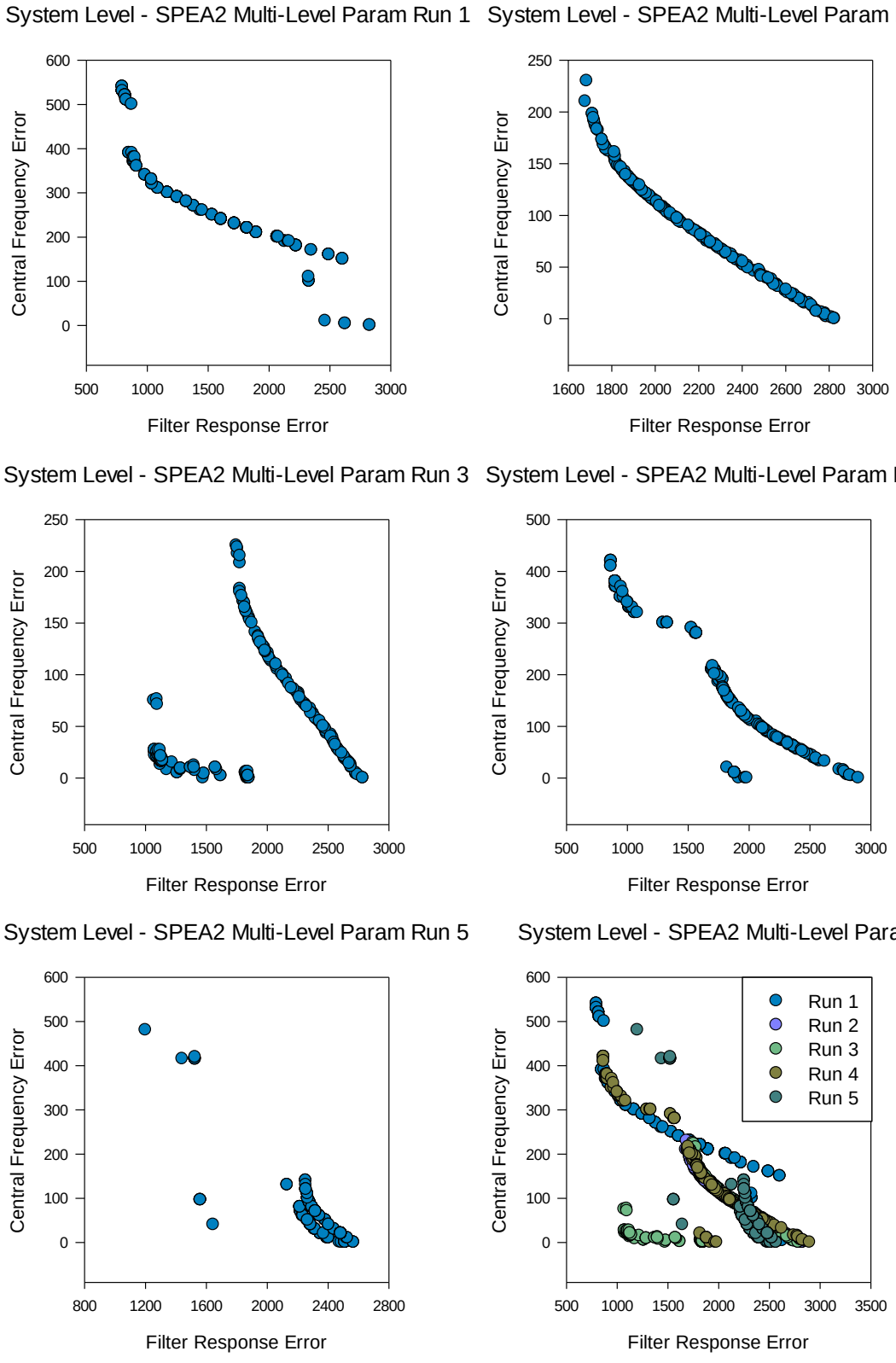


Figure C.10 System level run 1 – 5 final population sets (high + low) for SPEA2 multi-level parameterization

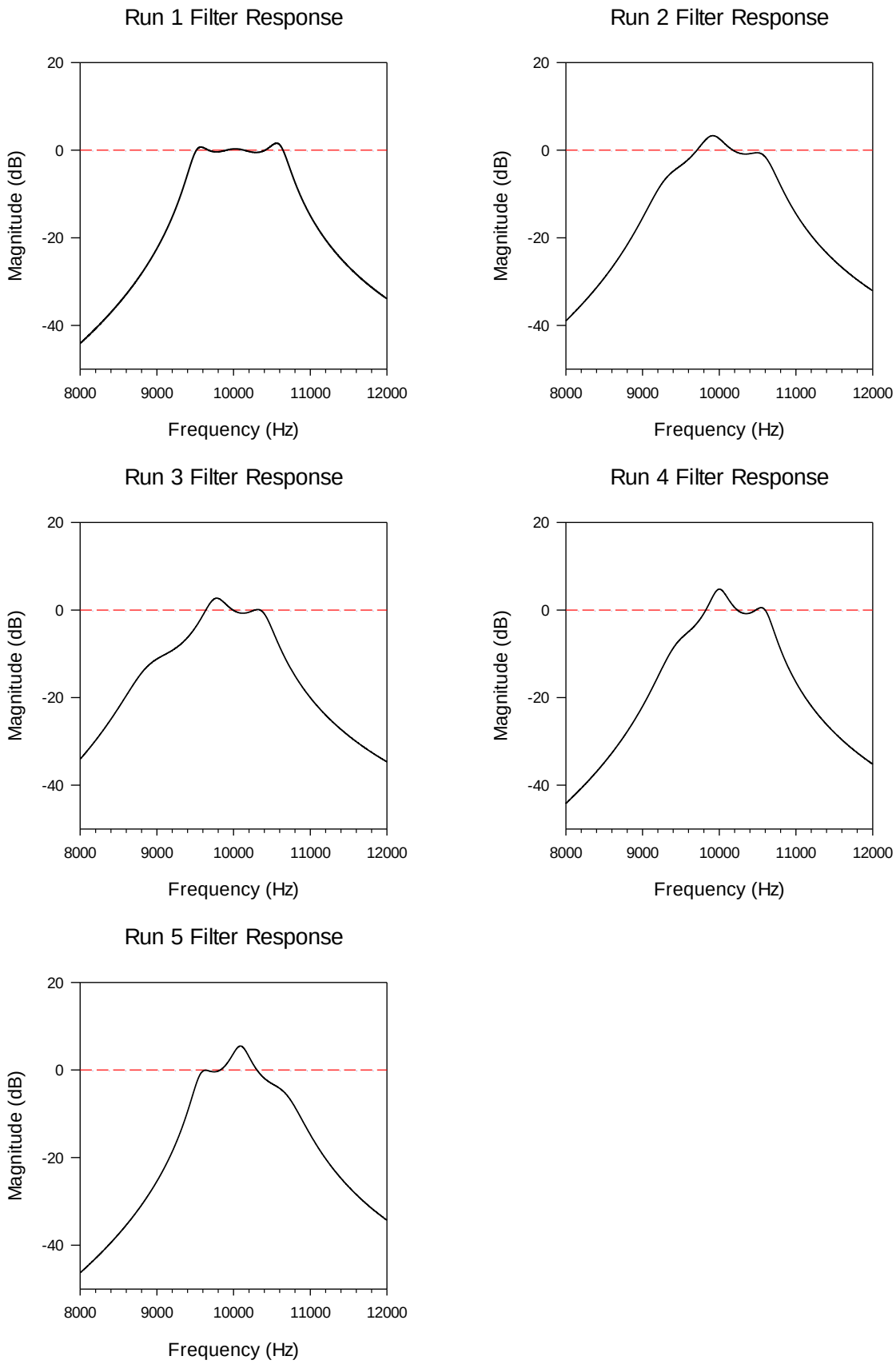


Figure C.11 System level run 1 – 5 best filter response ranked by filter frequency objective for NSGAII multi-level parameterization

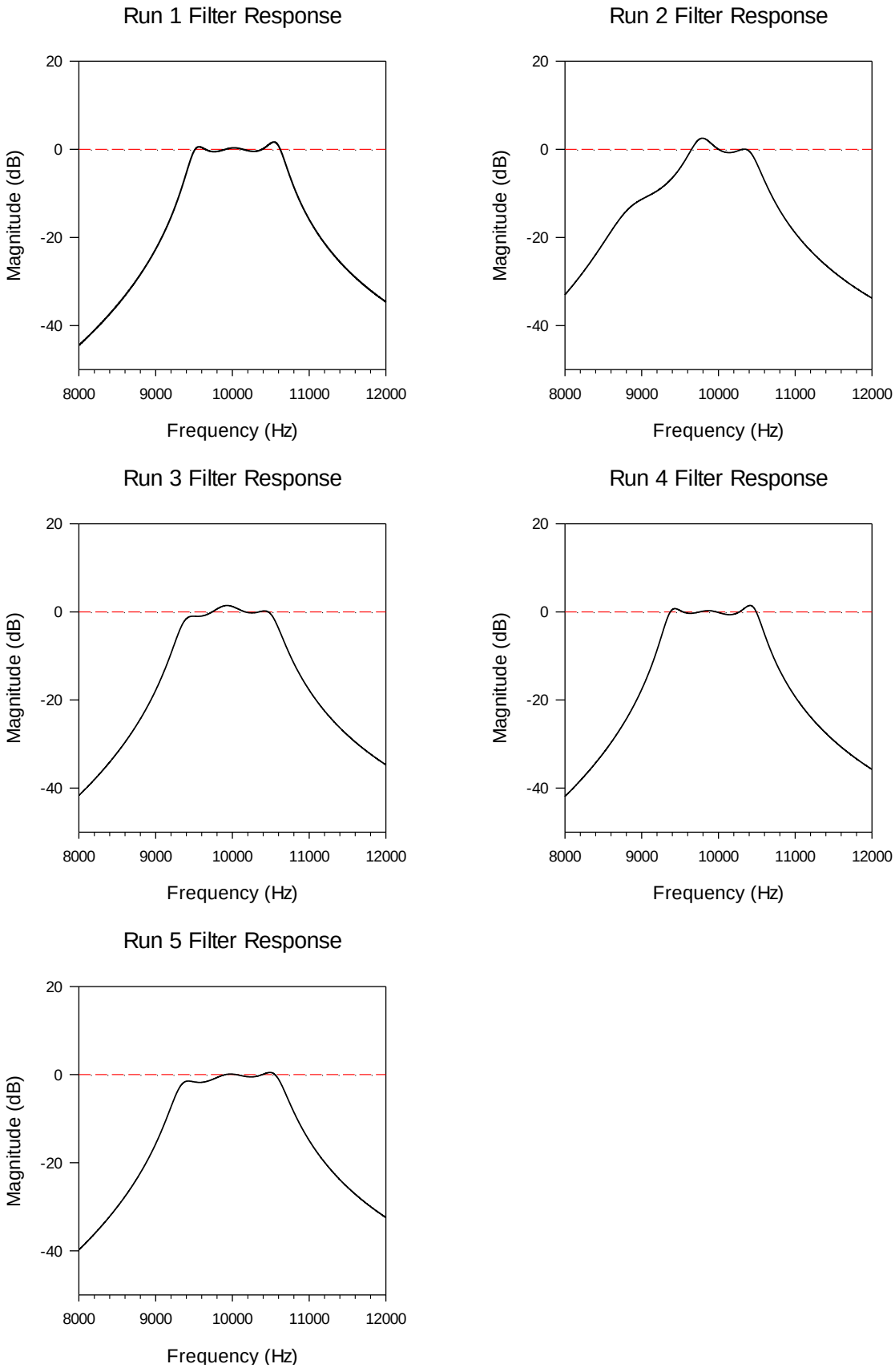


Figure C.12 System level run 1 – 5 best filter response ranked by filter frequency objective for SPEA2 multi-level parameterization

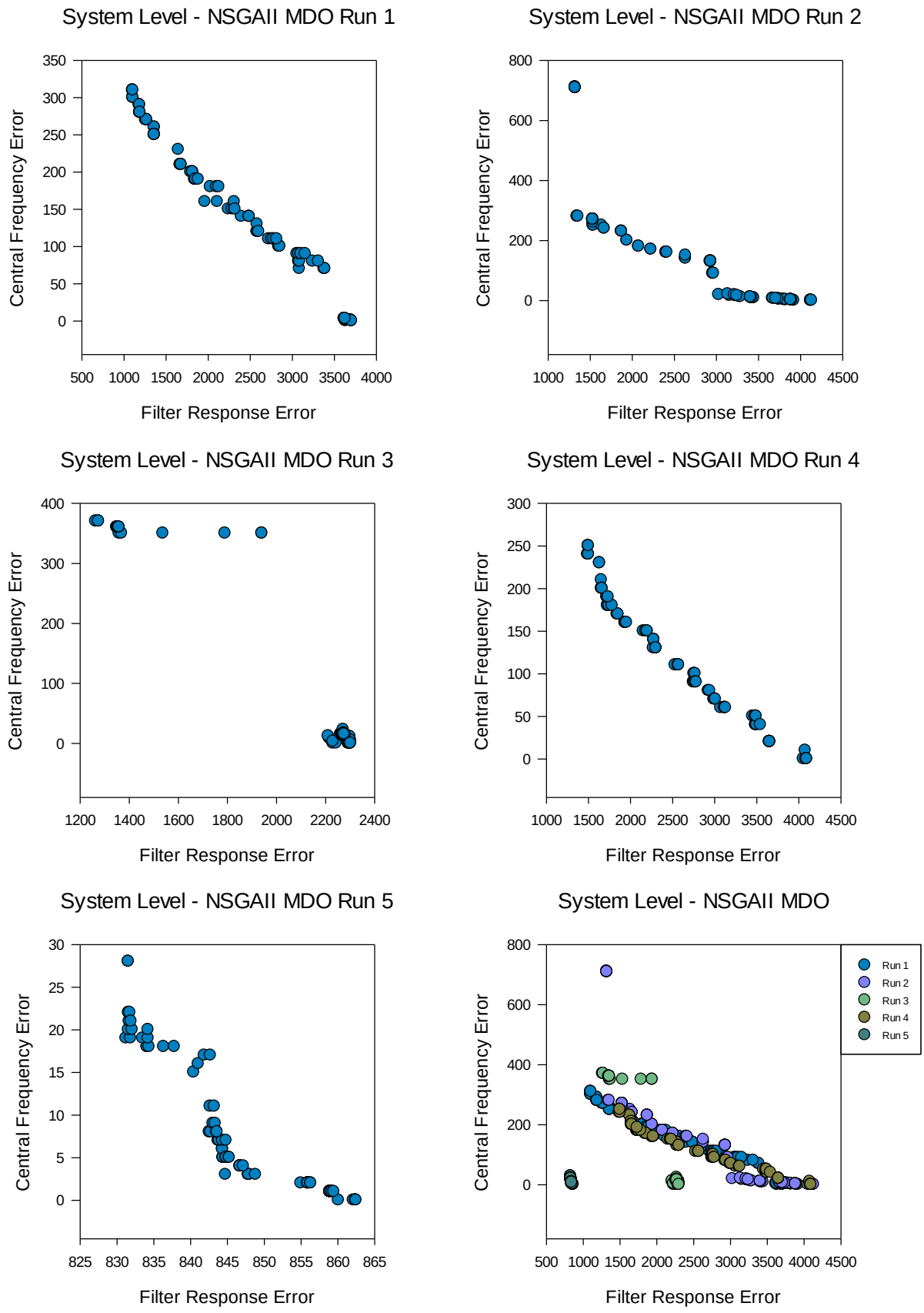


Figure C.13 System level run 1 – 5 final population sets for NSGAII multidisciplinary optimisation

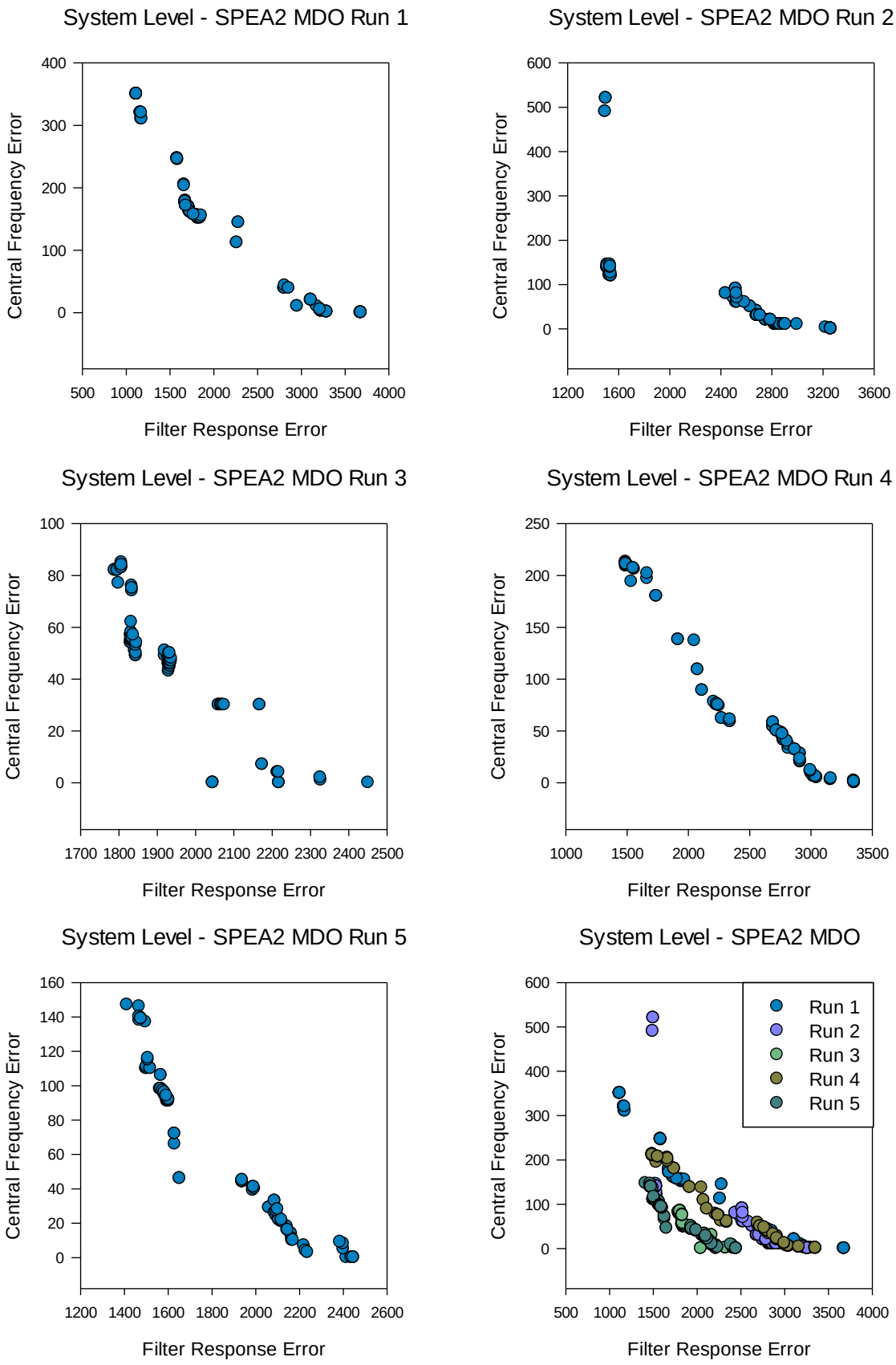


Figure C.14 System level run 1 – 5 final population sets for SPEA2 multidisciplinary optimisation

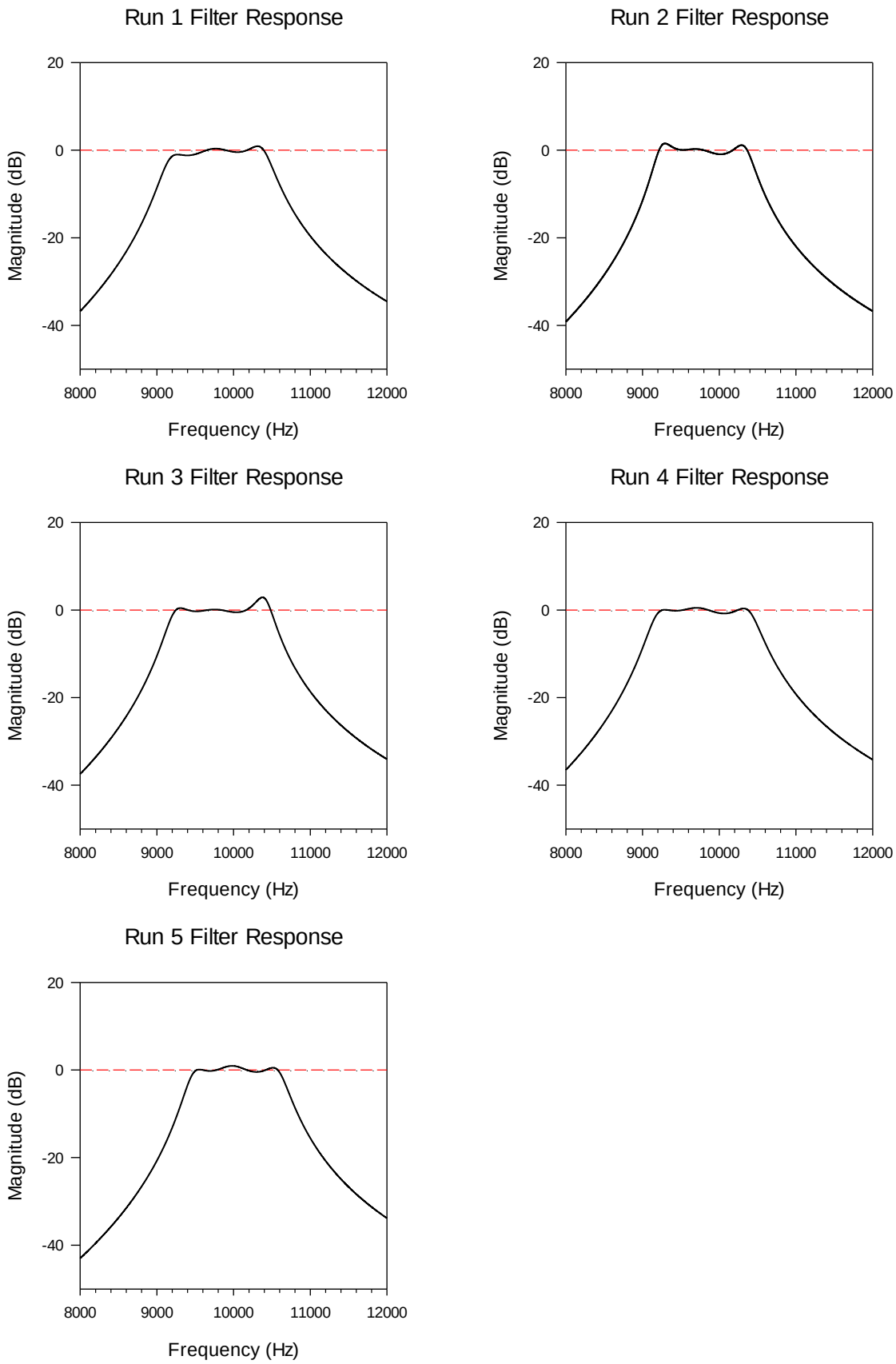


Figure C.15 System level run 1 – 5 best filter response ranked by filter frequency objective for NSGAII multidisciplinary optimisation

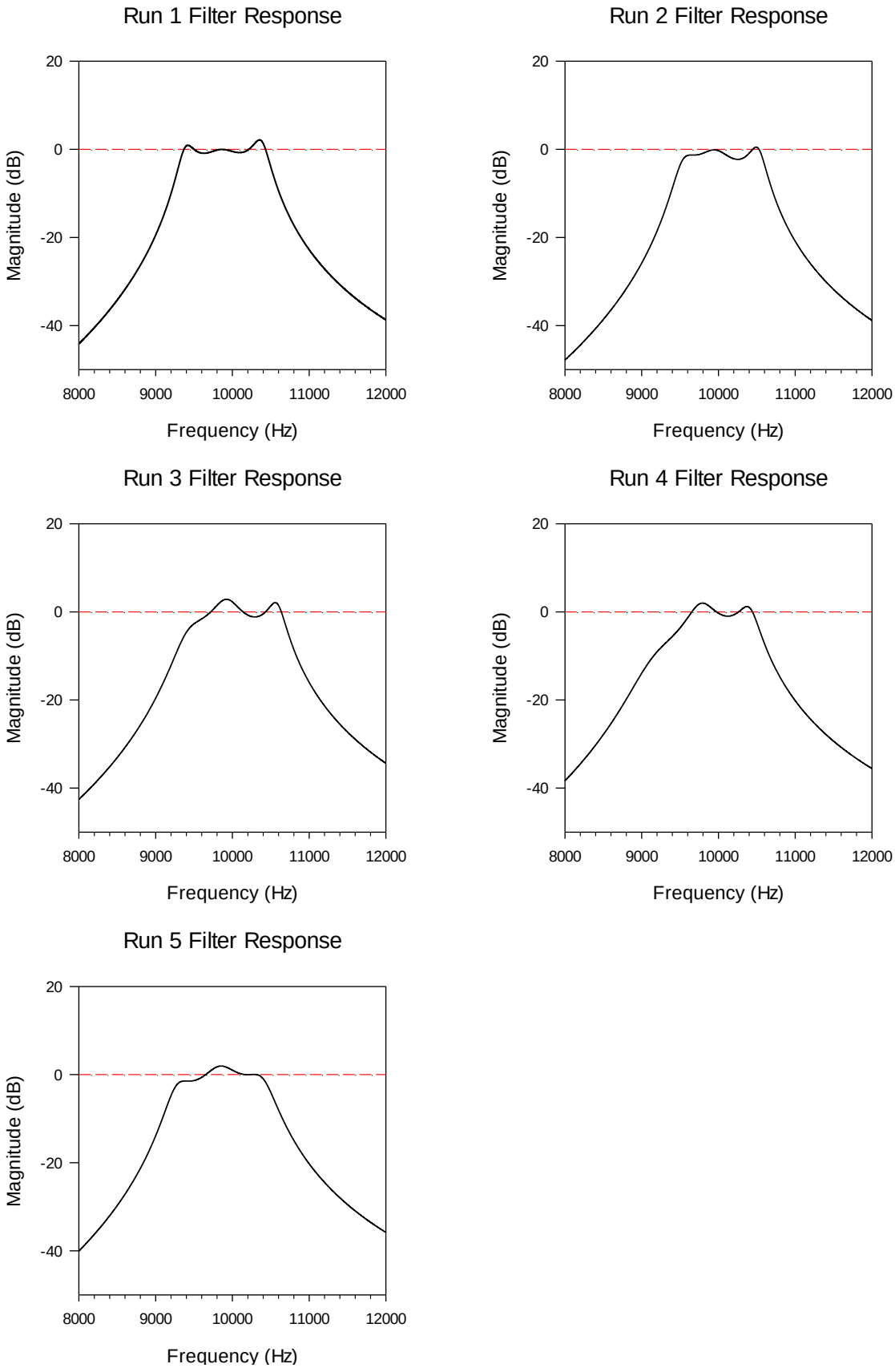
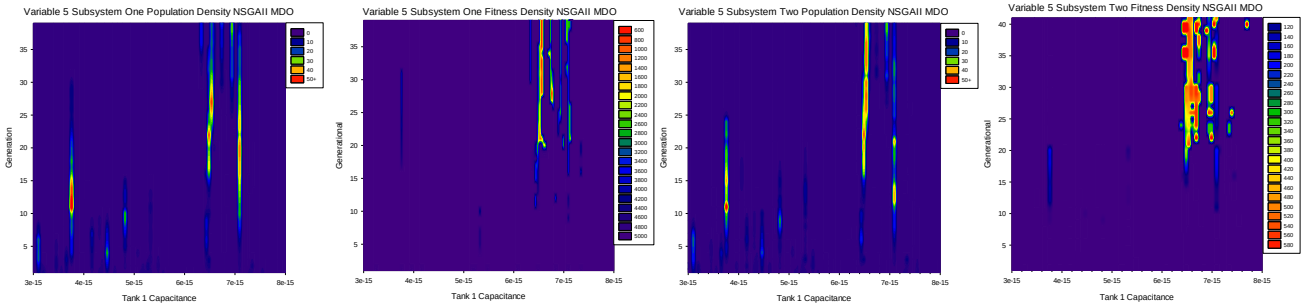
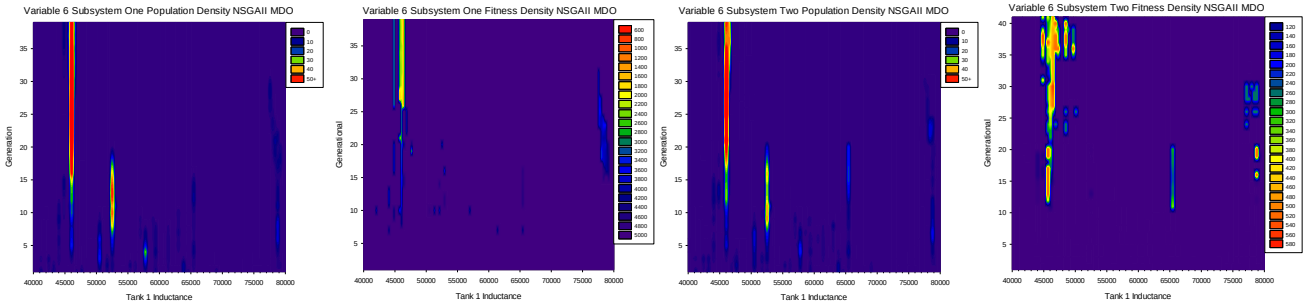
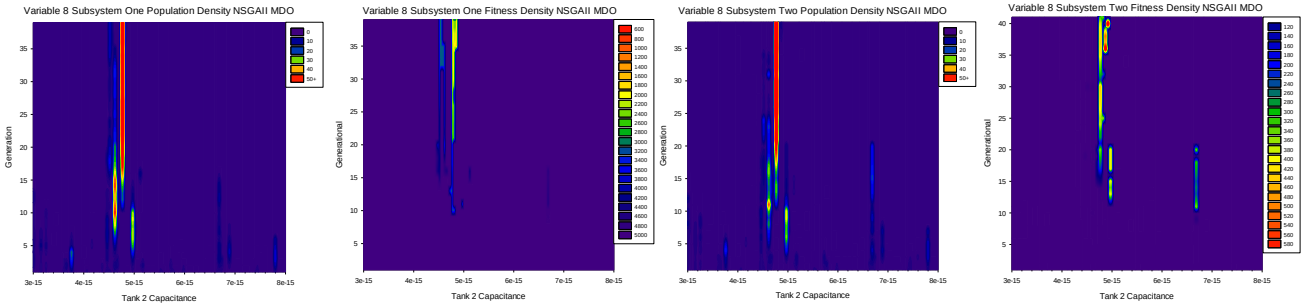
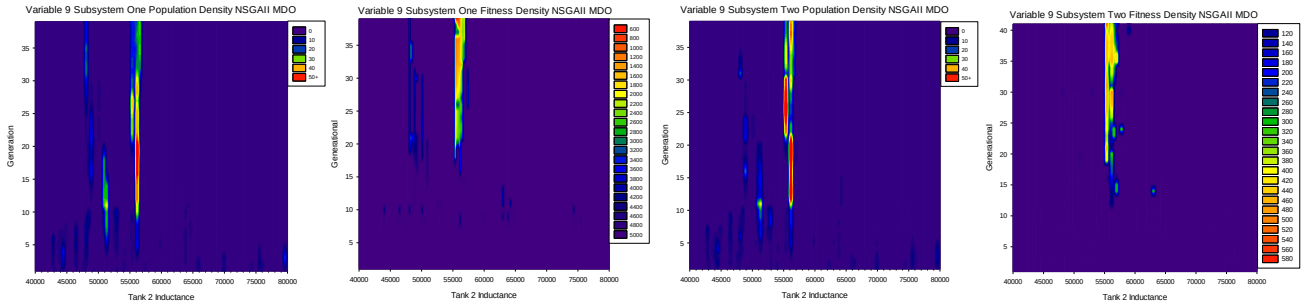


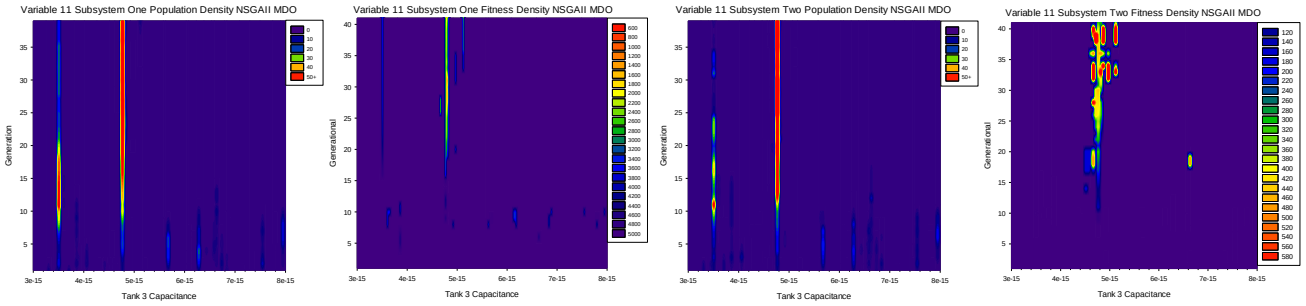
Figure C.15 System level run 1 – 5 best filter response ranked by filter frequency objective for SPEA2 multidisciplinary optimisation

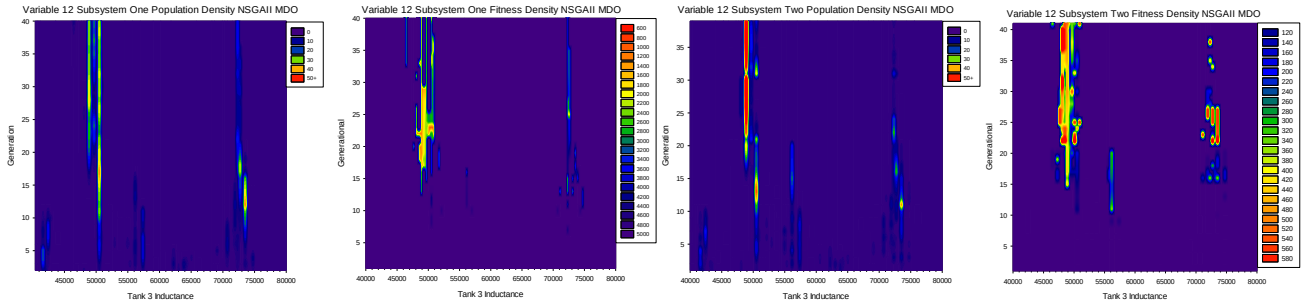

Variable 5 Subsystem One Population Density NSGAII MDO
Variable 5 Subsystem One Fitness Density NSGAII MDO
Variable 5 Subsystem Two Population Density NSGAII MDO
Variable 5 Subsystem Two Fitness Density NSGAII MDO
Generation
Generational
Tank 1 Capacitance


Variable 6 Subsystem One Population Density NSGAII MDO
Variable 6 Subsystem One Fitness Density NSGAII MDO
Variable 6 Subsystem Two Population Density NSGAII MDO
Variable 6 Subsystem Two Fitness Density NSGAII MDO
Generation
Generational
Tank 1 Inductance


Variable 8 Subsystem One Population Density NSGAII MDO
Variable 8 Subsystem One Fitness Density NSGAII MDO
Variable 8 Subsystem Two Population Density NSGAII MDO
Variable 8 Subsystem Two Fitness Density NSGAII MDO
Generation
Generational
Tank 2 Capacitance


Variable 9 Subsystem One Population Density NSGAII MDO
Variable 9 Subsystem One Fitness Density NSGAII MDO
Variable 9 Subsystem Two Population Density NSGAII MDO
Variable 9 Subsystem Two Fitness Density NSGAII MDO
Generation
Generational
Tank 2 Inductance


Variable 11 Subsystem One Population Density NSGAII MDO
Variable 11 Subsystem One Fitness Density NSGAII MDO
Variable 11 Subsystem Two Population Density NSGAII MDO
Variable 11 Subsystem Two Fitness Density NSGAII MDO
Generation
Generational
Tank 3 Capacitance


Variable 12 Subsystem One Population Density NSGAII MDO
Variable 12 Subsystem One Fitness Density NSGAII MDO
Variable 12 Subsystem Two Population Density NSGAII MDO
Variable 12 Subsystem Two Fitness Density NSGAII MDO
Generation
Generational
Tank 3 Inductance

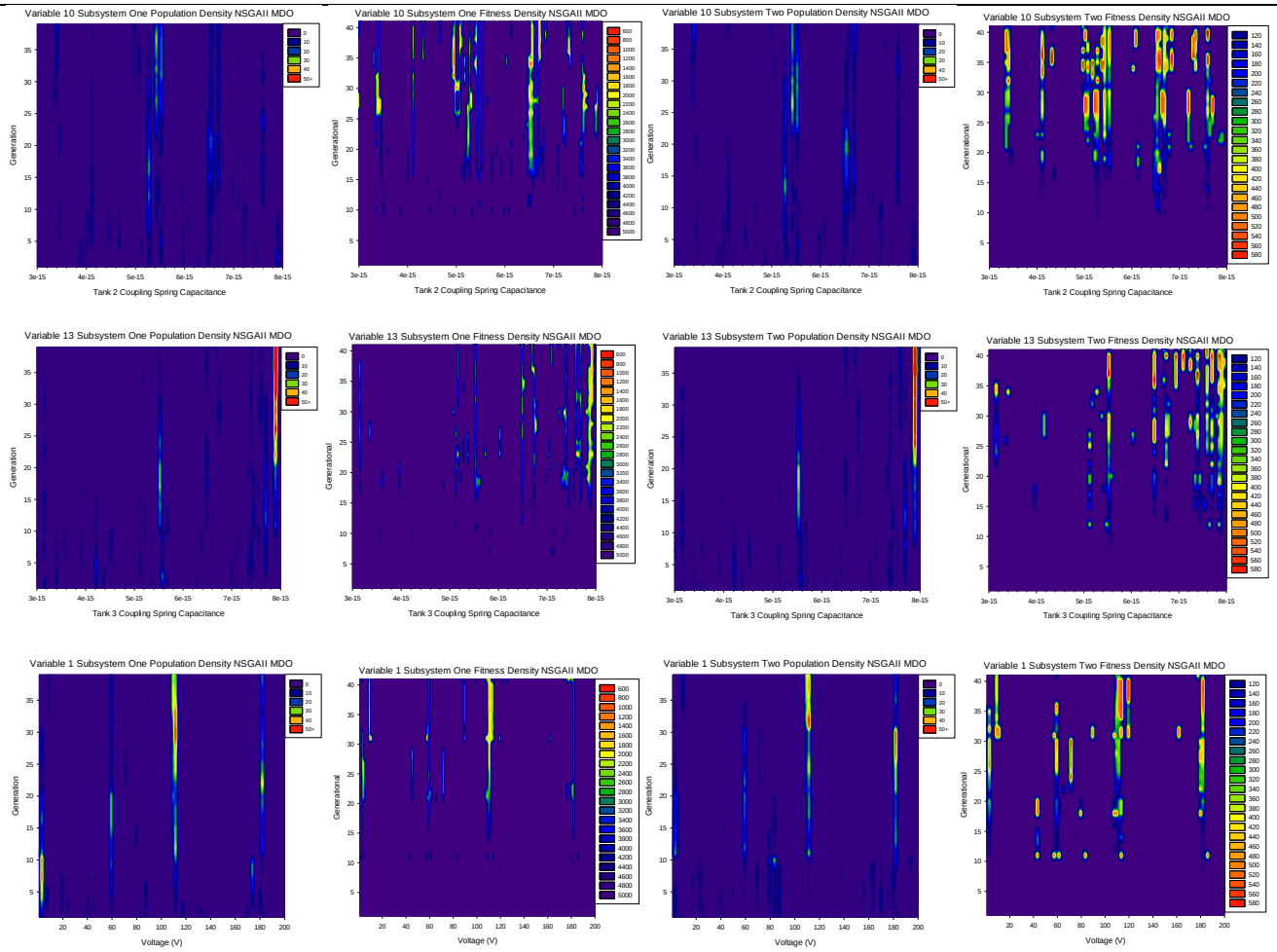


Figure C.16 Generational histogram plots for system level multidisciplinary optimisation variables plotted against population density and population fitness for both subsystem one and two sets of NSGAII run 1.

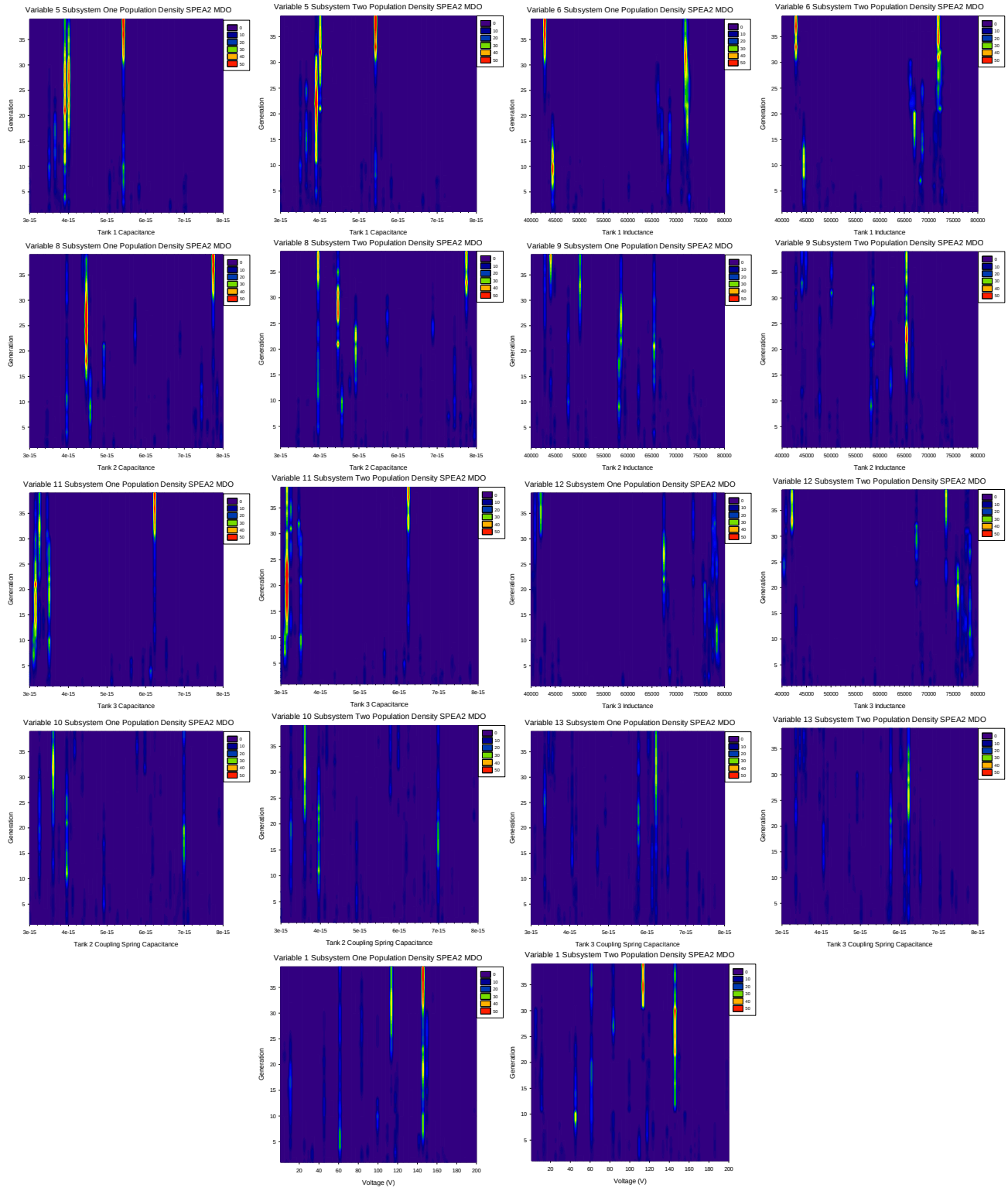


Figure C.17 Generational histogram plots for system level multidisciplinary optimisation variables plotted against population density for both subsystem one and two sets of SPEA2 run 1

C.2 Device level experimental results

The following section holds device level results for the single level, multi-level evaluation, multi-level parameterization and multidisciplinary optimisation strategies, consisting of the final population sets for each strategy.

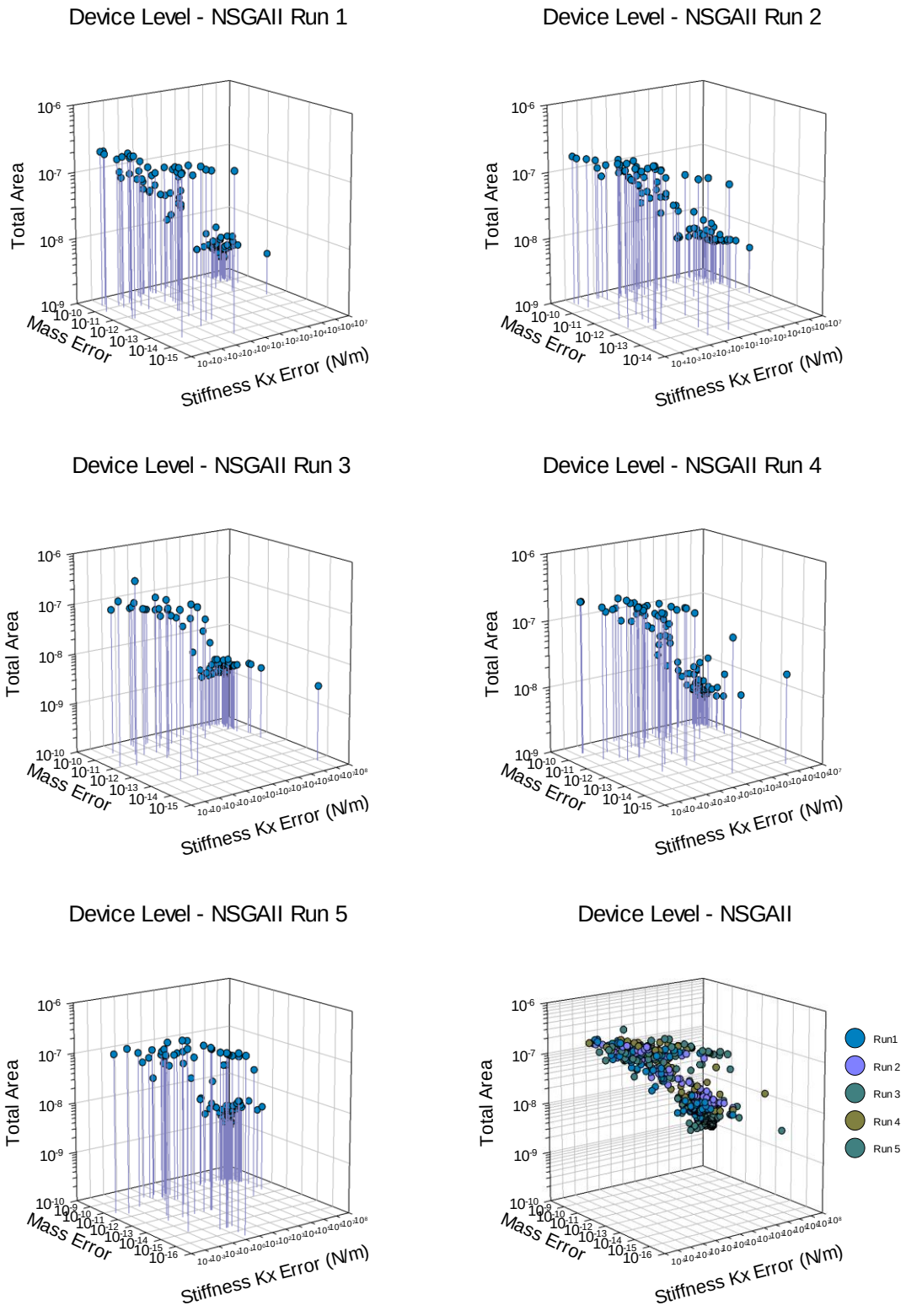


Figure C.18 Device level run 1 – 5 final population sets for NSGAII

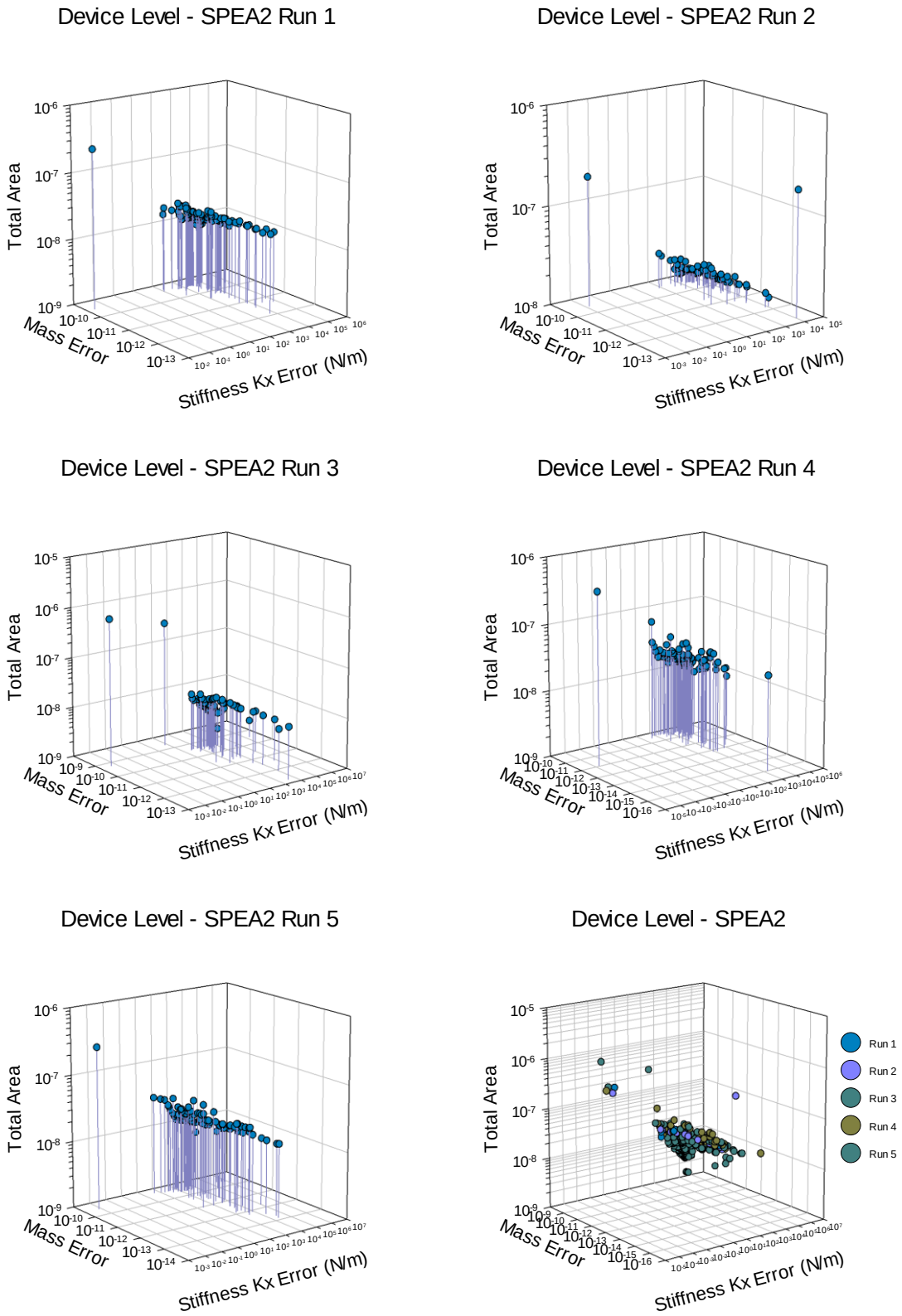


Figure C.19 Device level run 1 – 5 final population sets for SPEA2

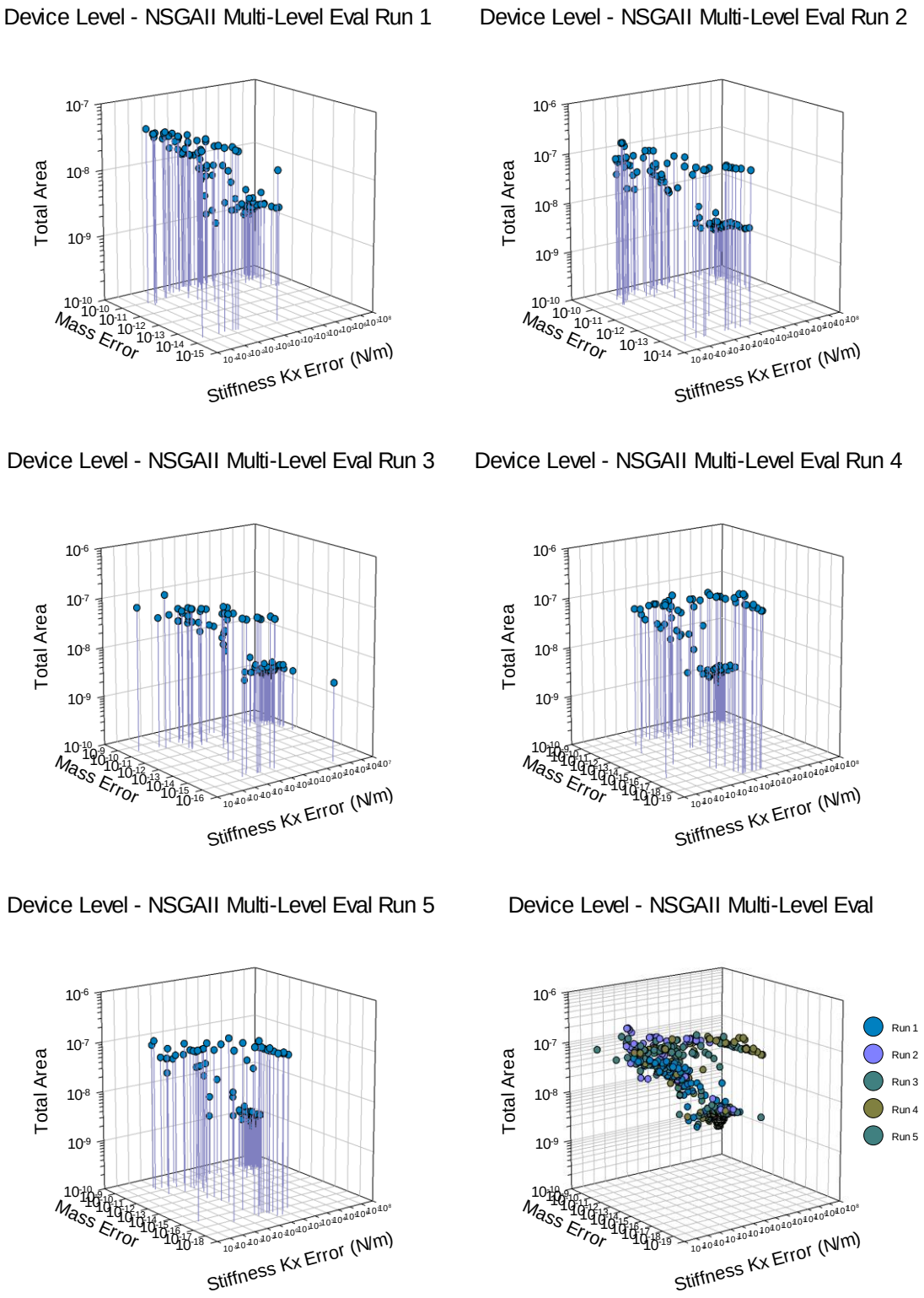


Figure C.20 Device level run 1 – 5 final population sets for NSGAII multi-level evaluation

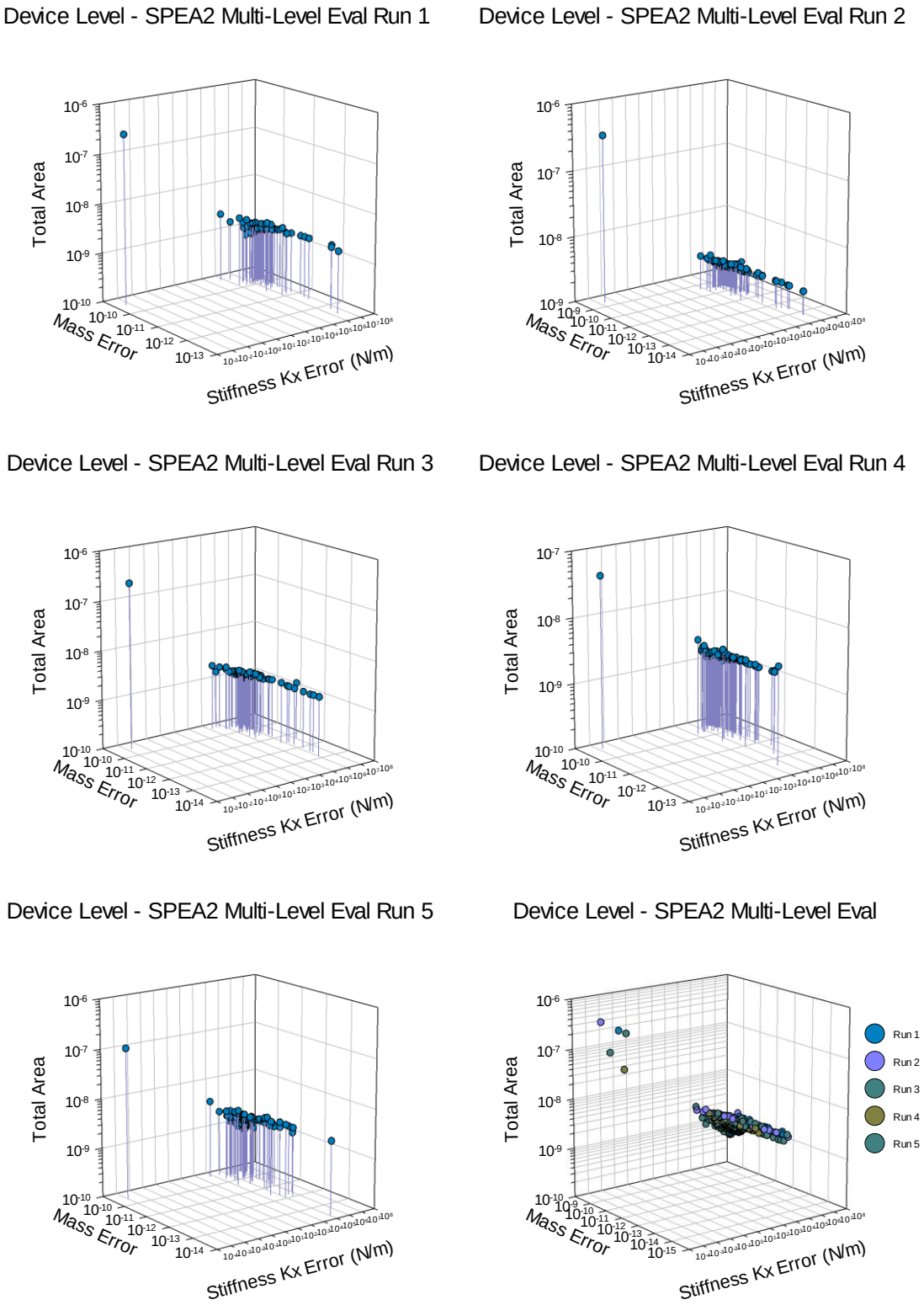


Figure C.21 Device level run 1 – 5 final population sets for SPEA2 multi-level evaluation

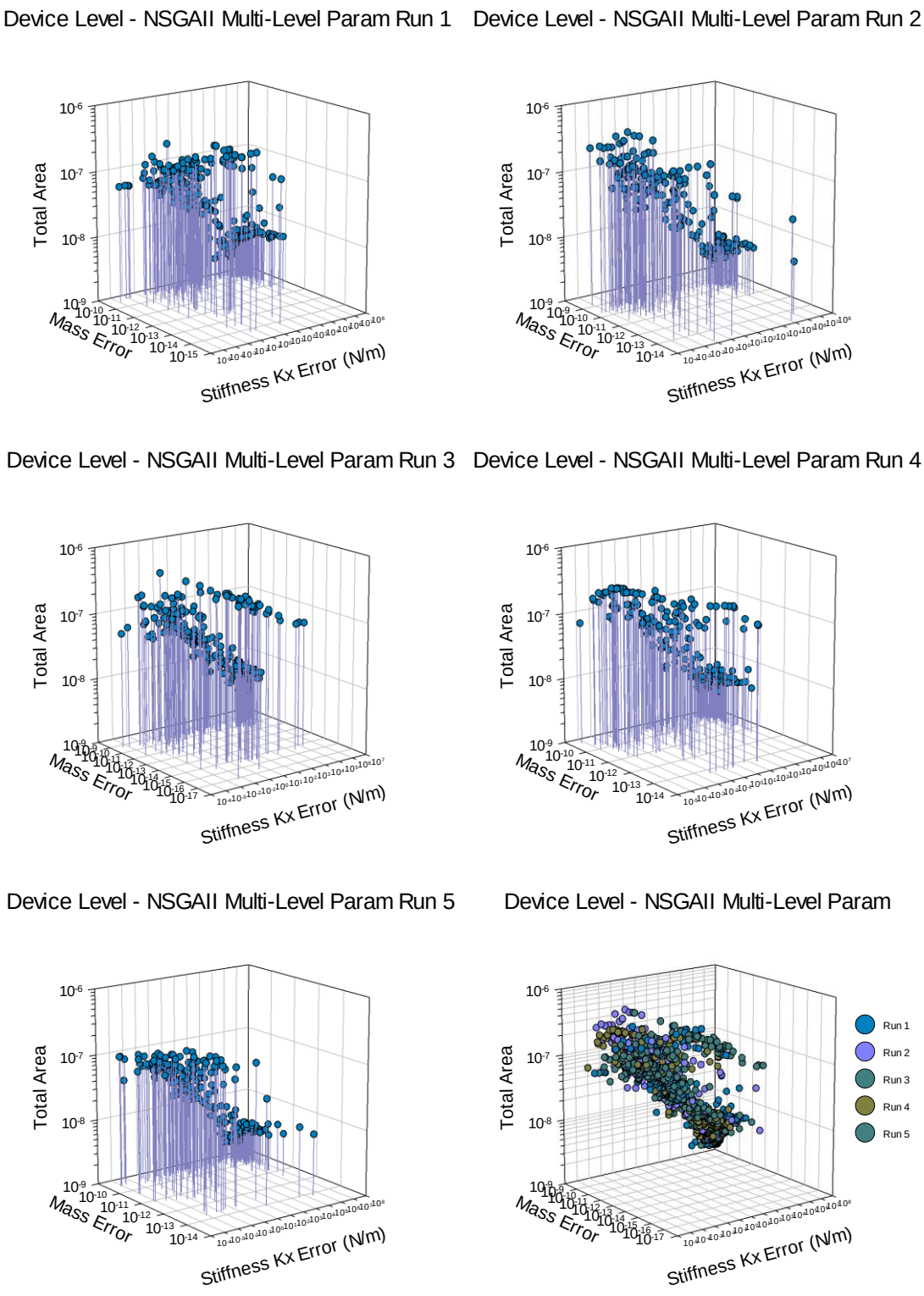


Figure C.21 Device level run 1 – 5 final population sets (high + low) for NSGAII multi-level parameterization

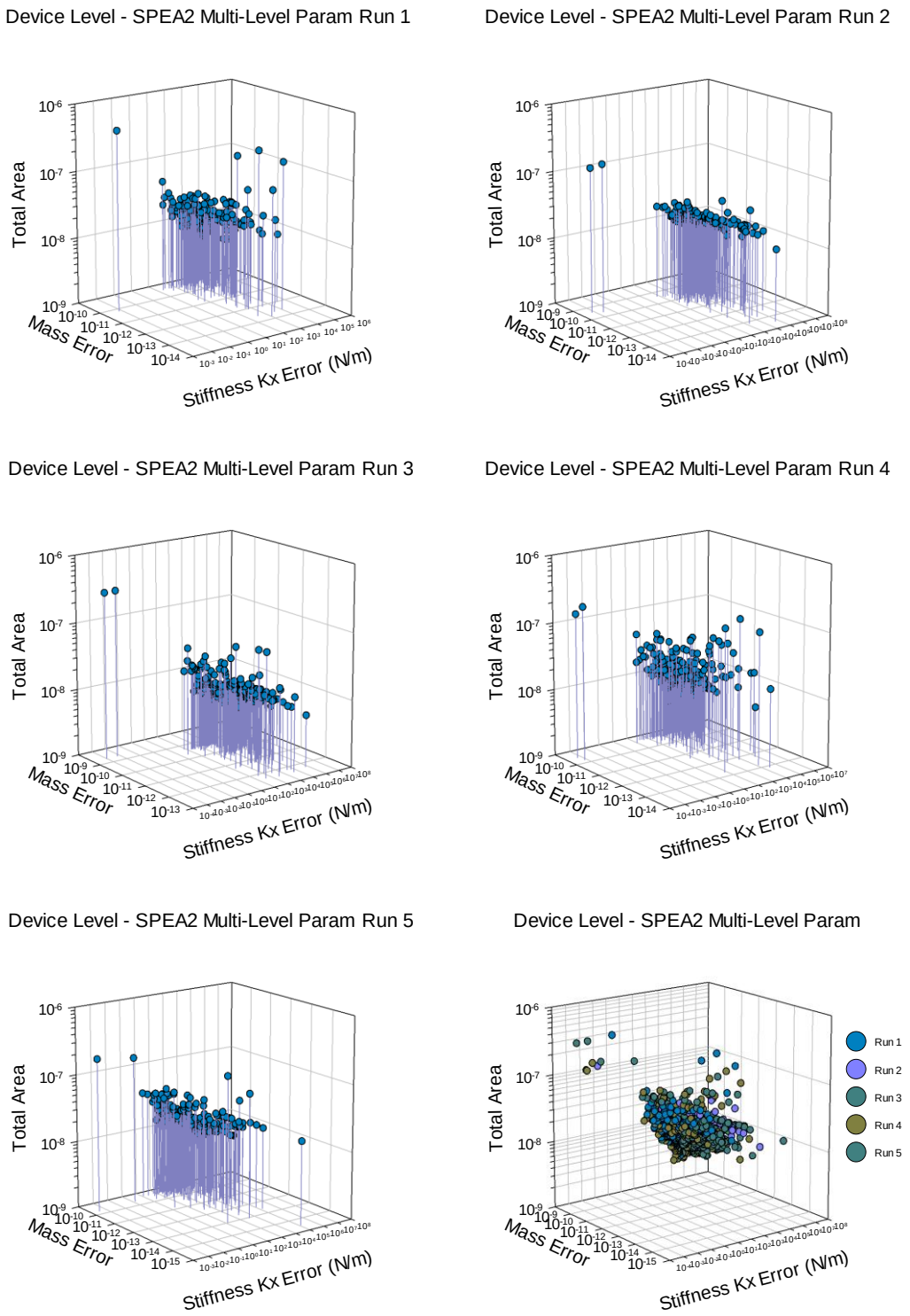


Figure C.22 Device level run 1 – 5 final population sets (high + low) for SPEA2 multi-level parameterization

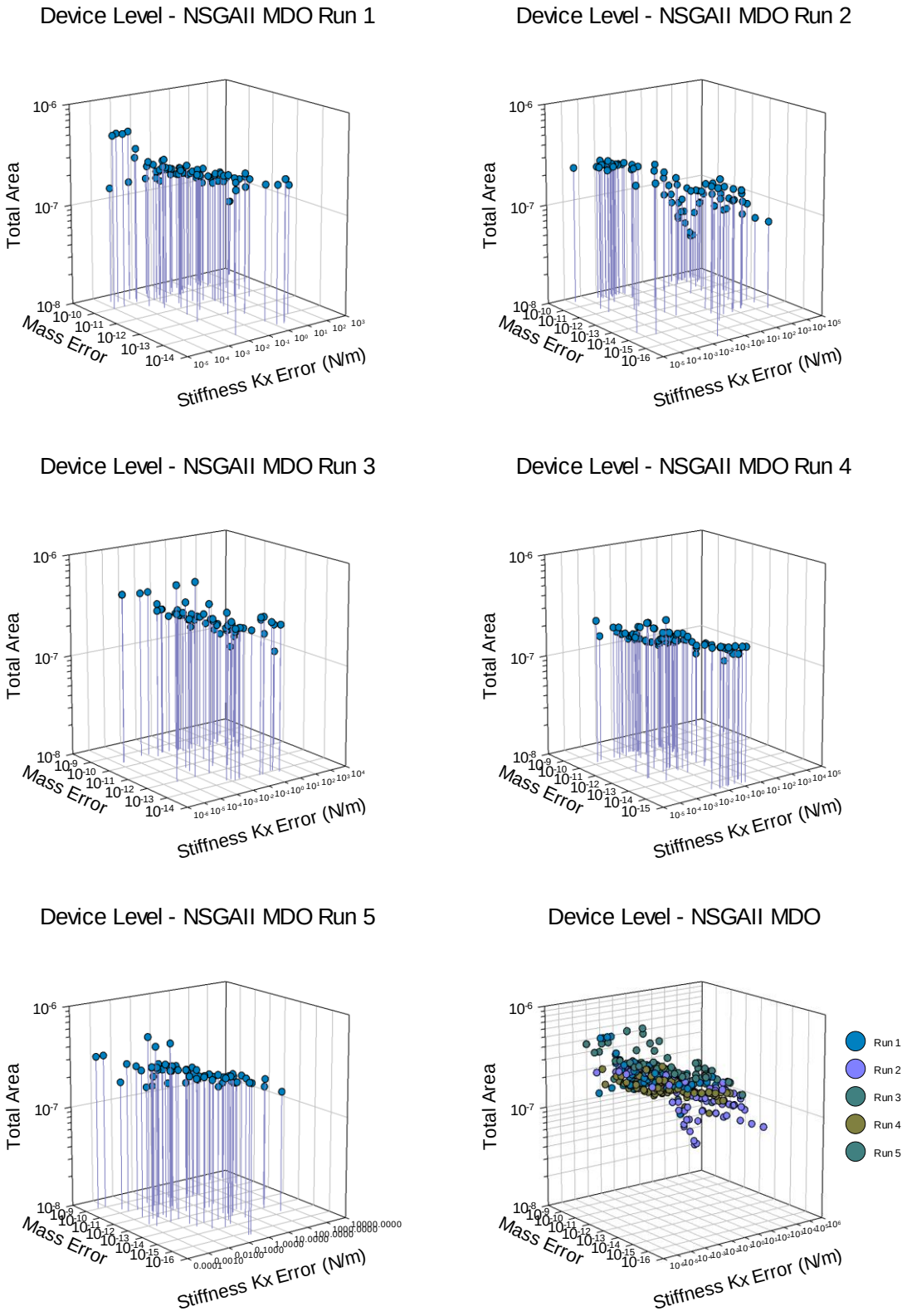


Figure C.23 Device level run 1 – 5 final population sets for NSGAII multidisciplinary optimisation

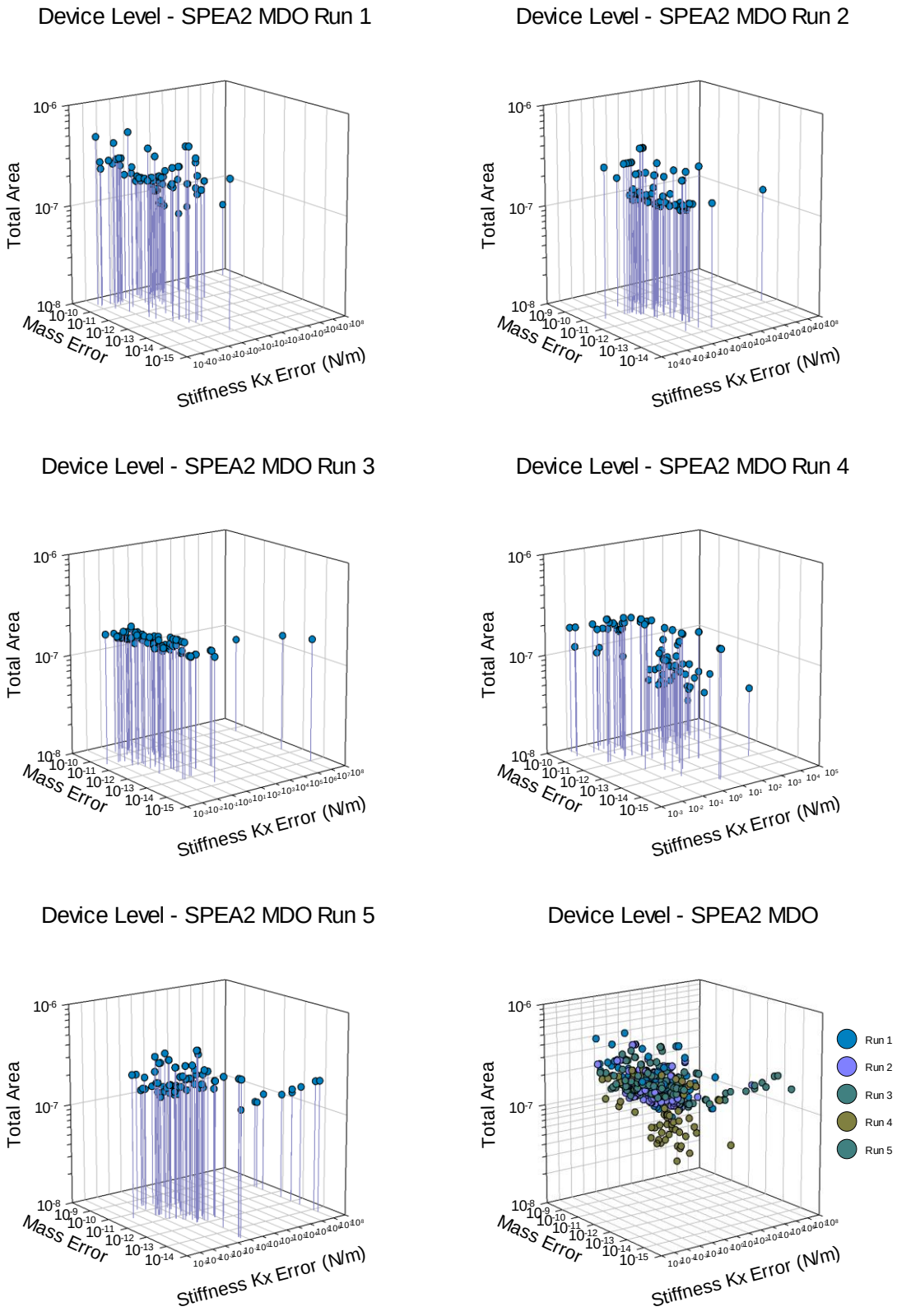


Figure C.24 Device level run 1 – 5 final population sets for SPEA2 multidisciplinary optimisation

C.3 Physical level experimental results

The following section holds physical level results for the single level, multi-level evaluation, multi-level parameterization and multidisciplinary optimisation strategies, consisting of the final population sets for each strategy.

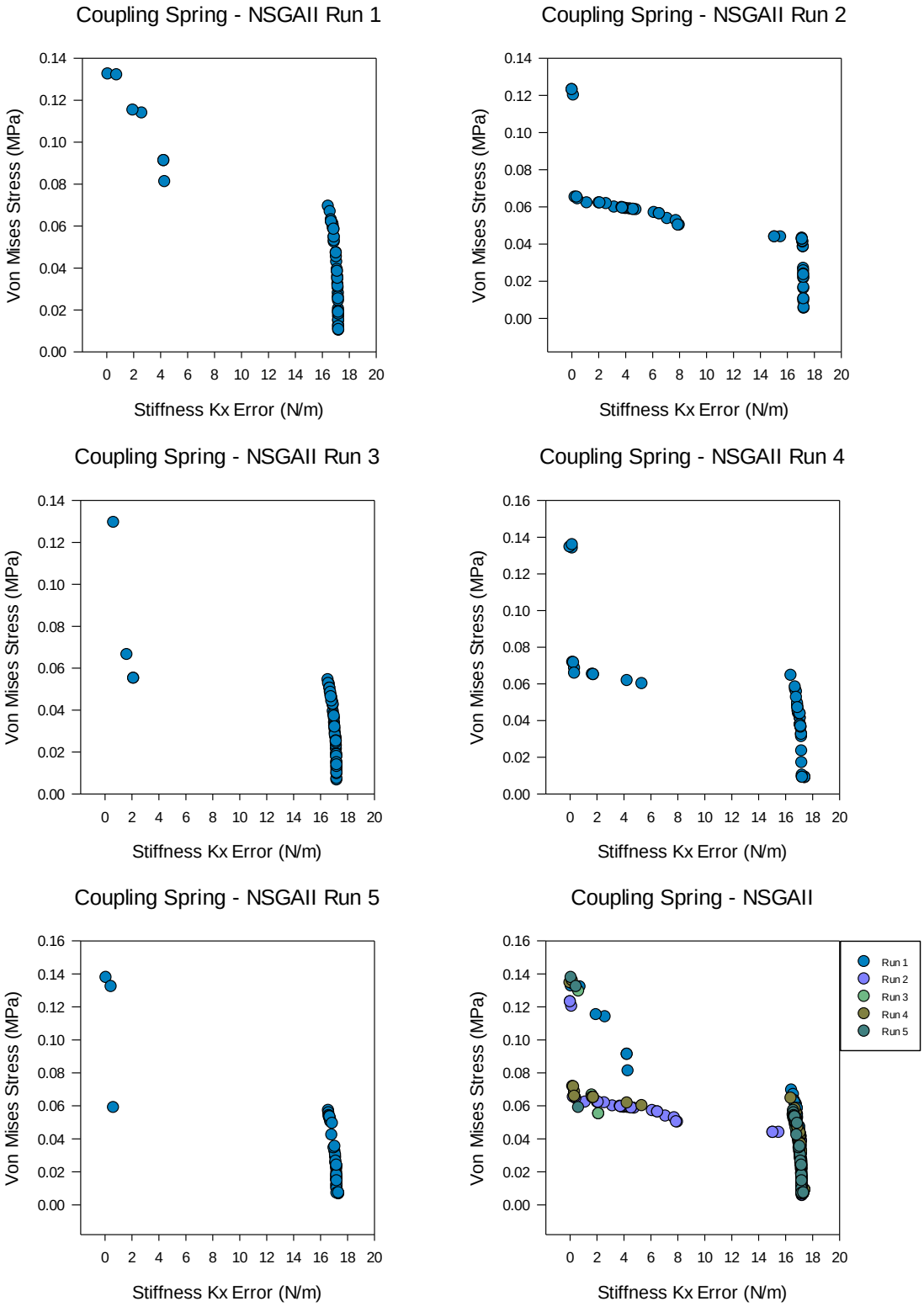


Figure C.25 Physical level run 1 – 5 final population sets for NSGAII

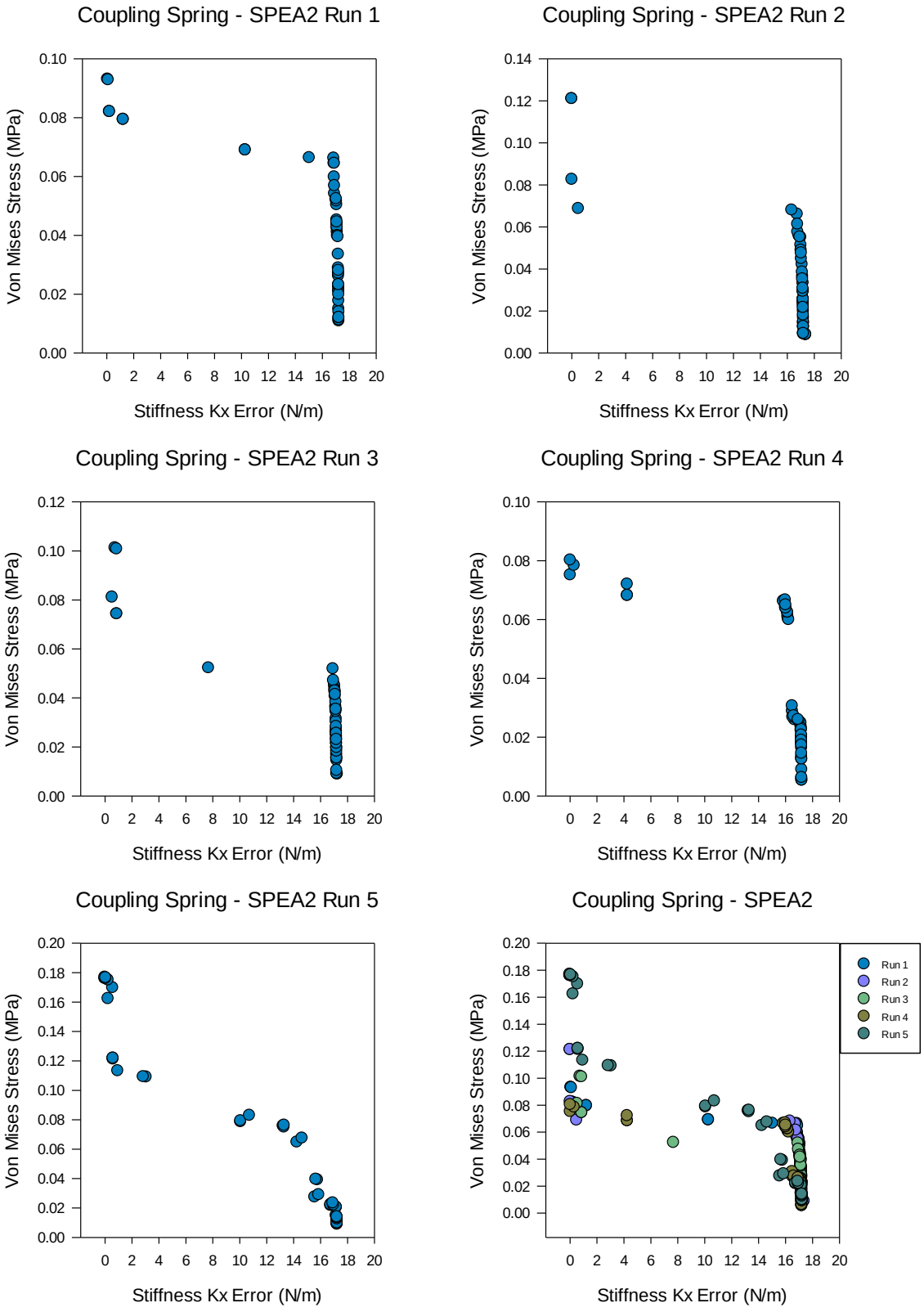


Figure C.26 Physical level run 1 – 5 final population sets for SPEA2

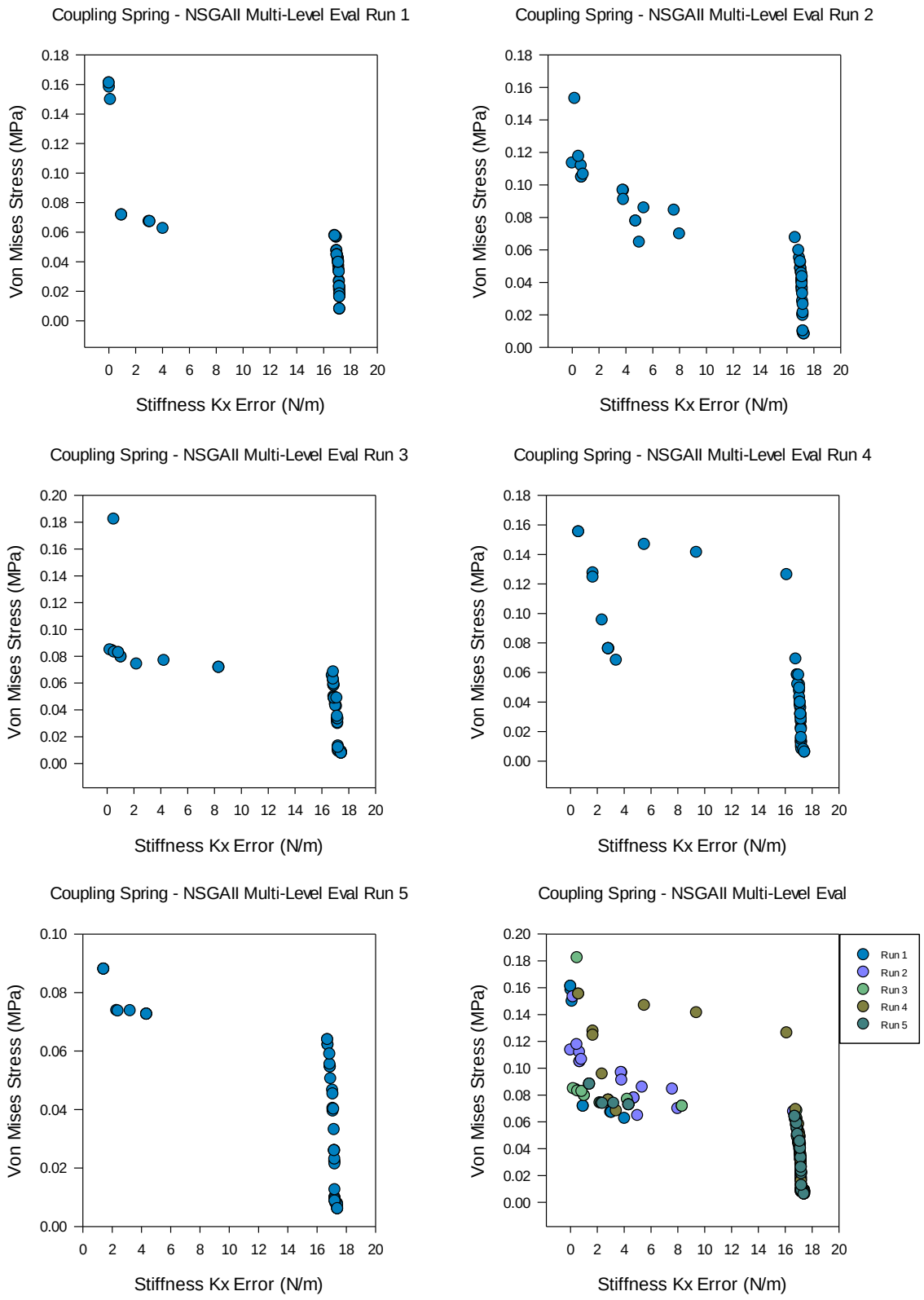


Figure C.27 Physical level run 1 – 5 final population sets for NSGAII multi-level evaluation

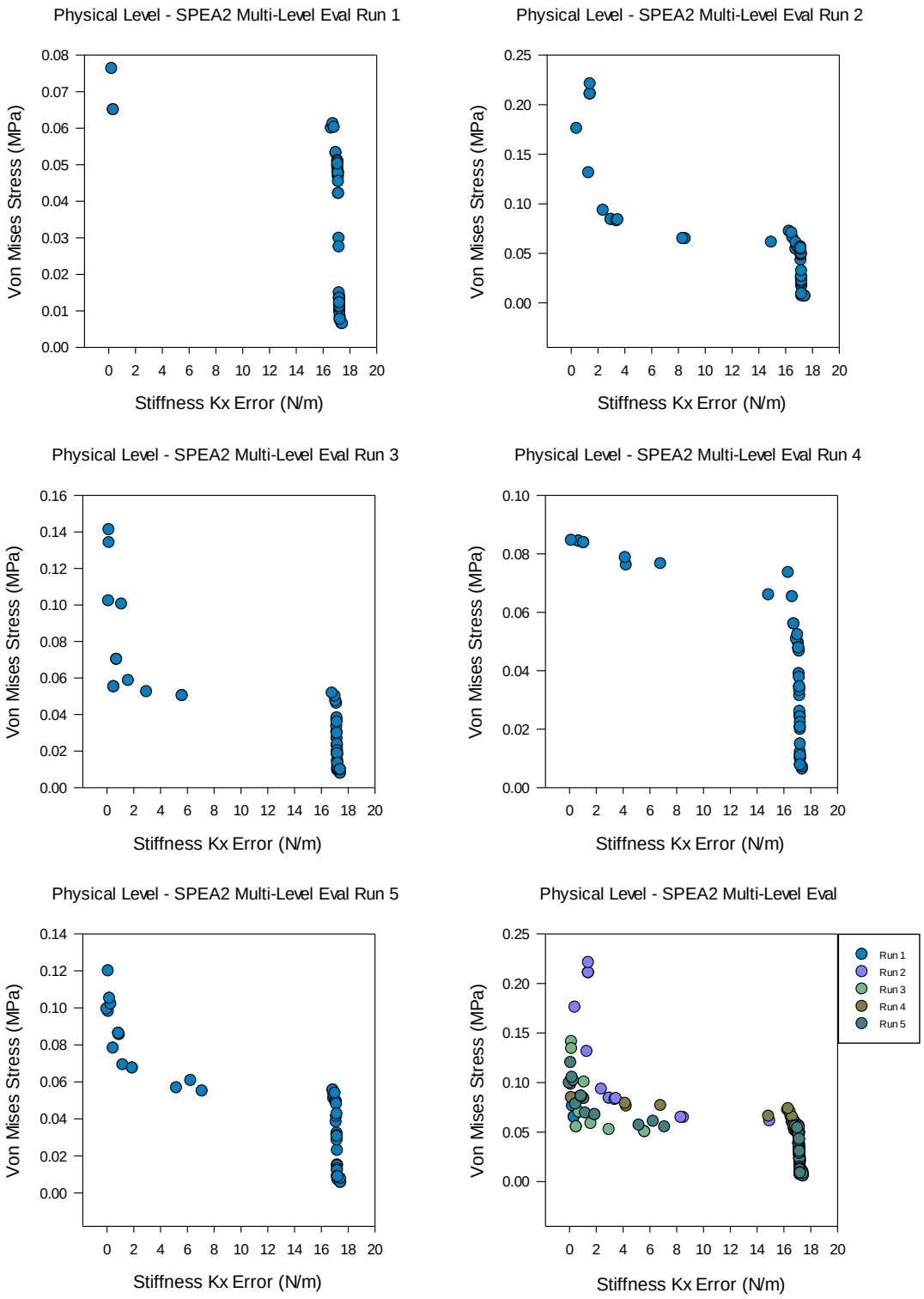


Figure C.28 Physical level run 1 – 5 final population sets for SPEA2 multi-level evaluation

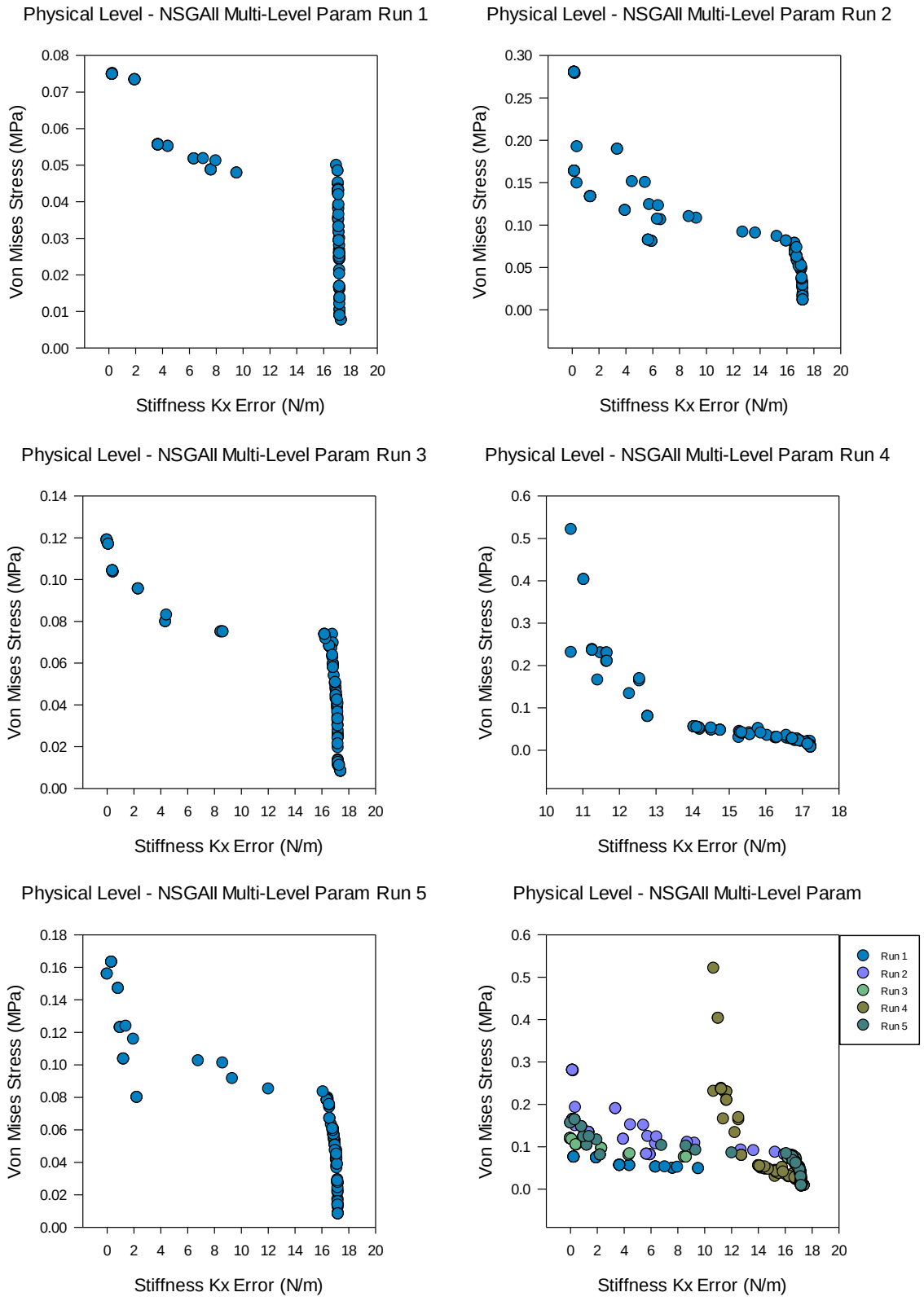


Figure C.29 Physical level run 1 – 5 final population sets for NSGAII multi-level parameterization

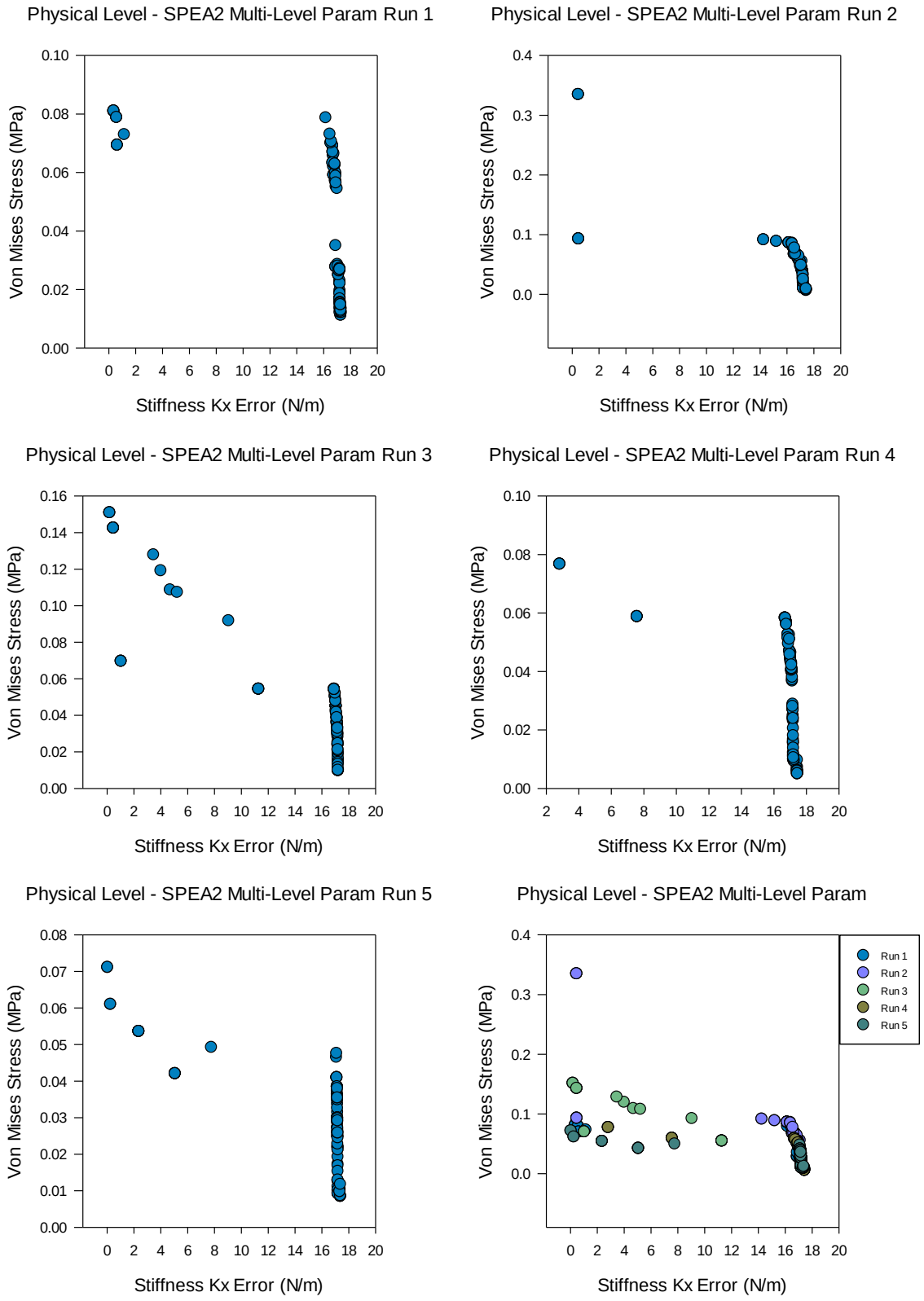


Figure C.30 Physical level run 1 – 5 final population sets for NSGAII multi-level parameterization

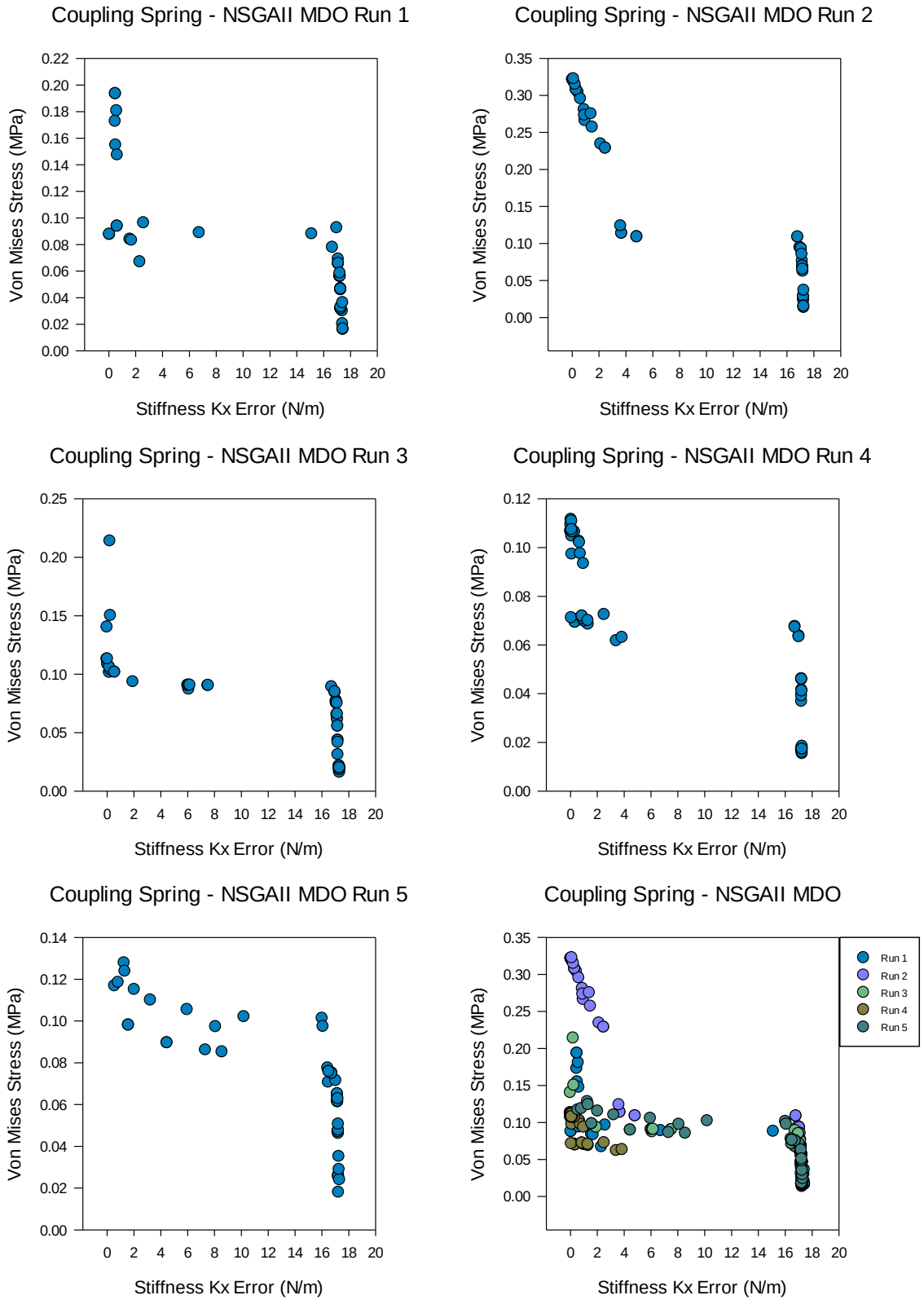


Figure C.31 Physical level run 1 – 5 final population sets for NSGAII multidisciplinary optimisation

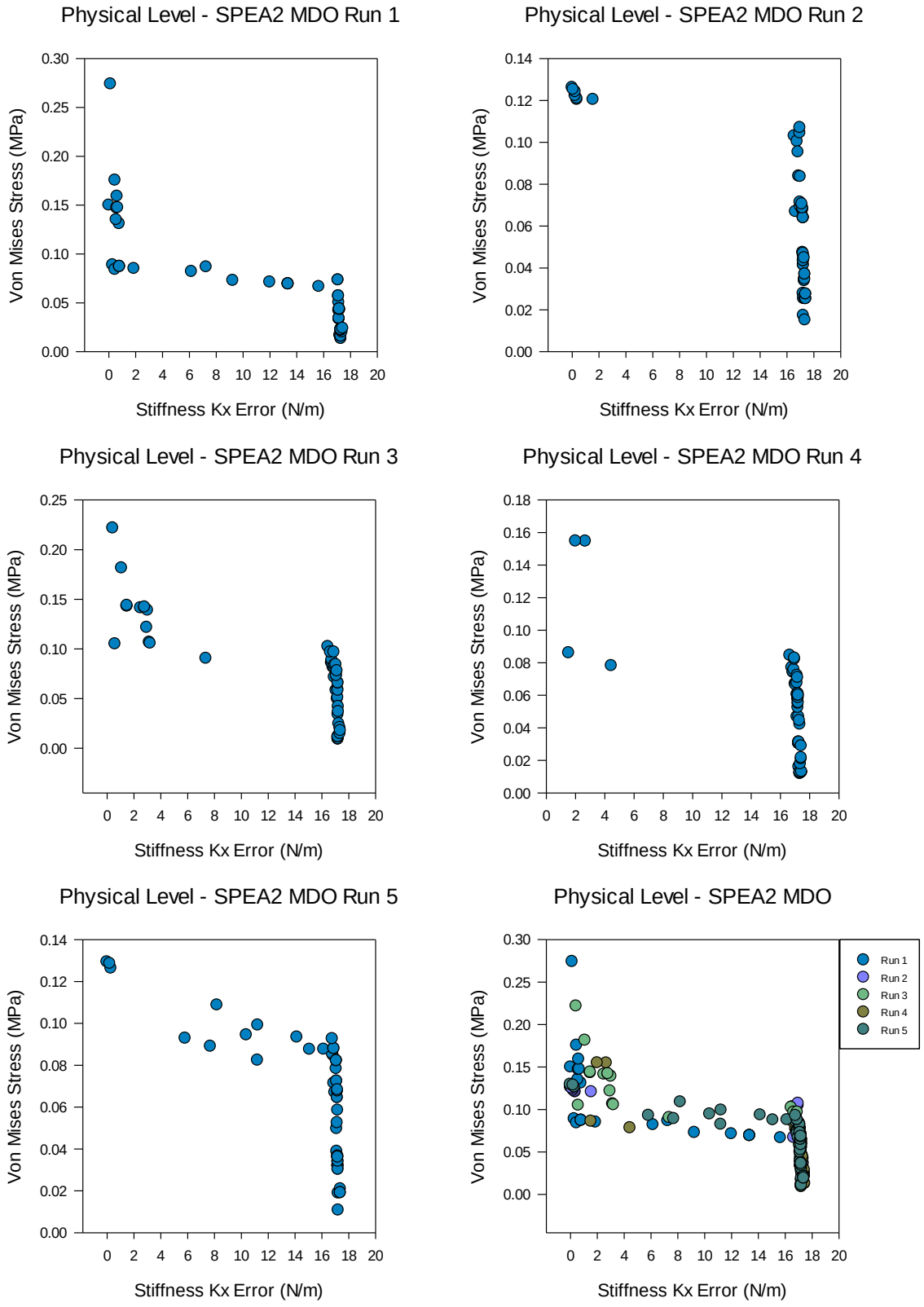


Figure C.32 Physical level run 1 – 5 final population sets for SPEA2 multidisciplinary optimisation

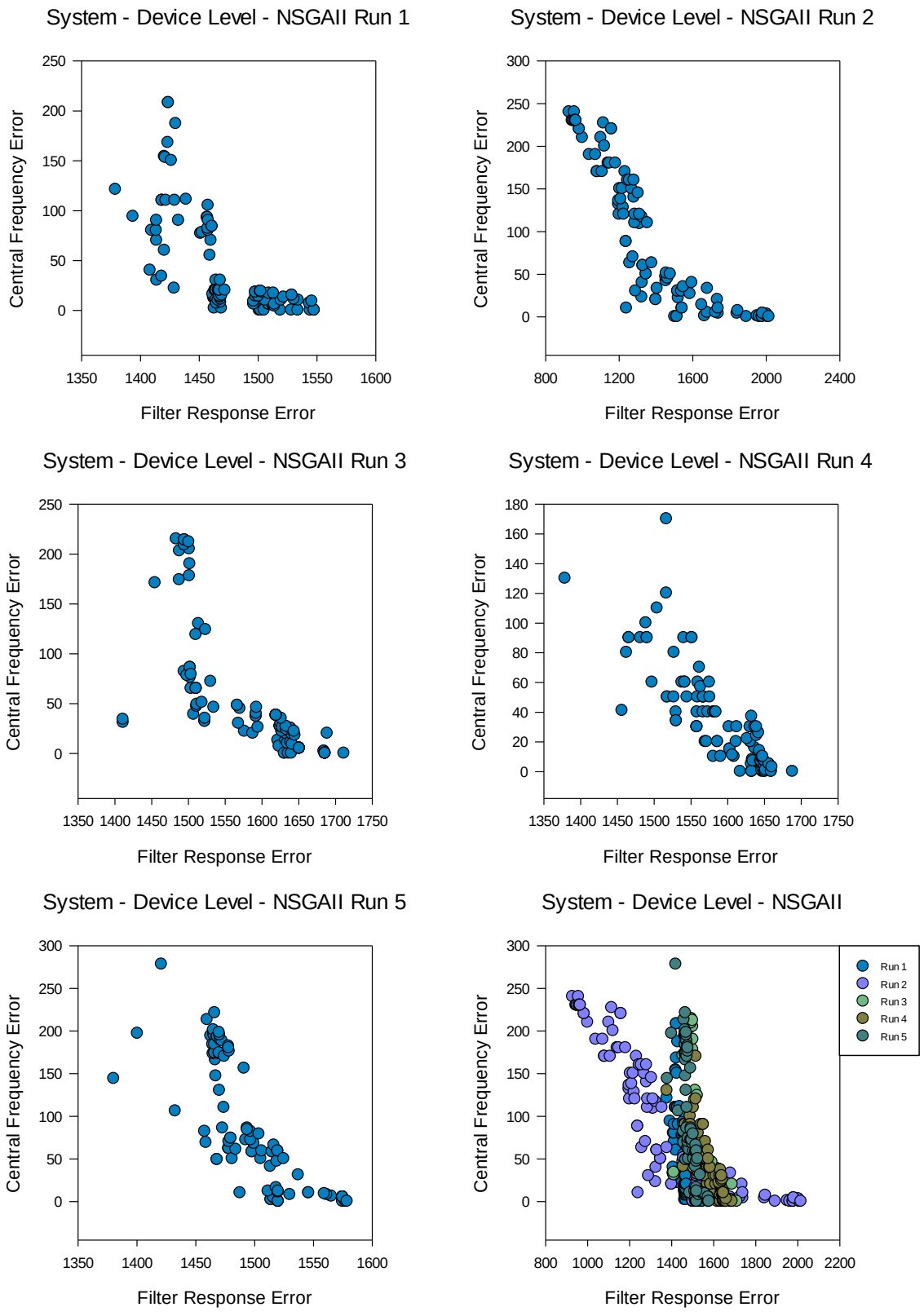
D

Appendix

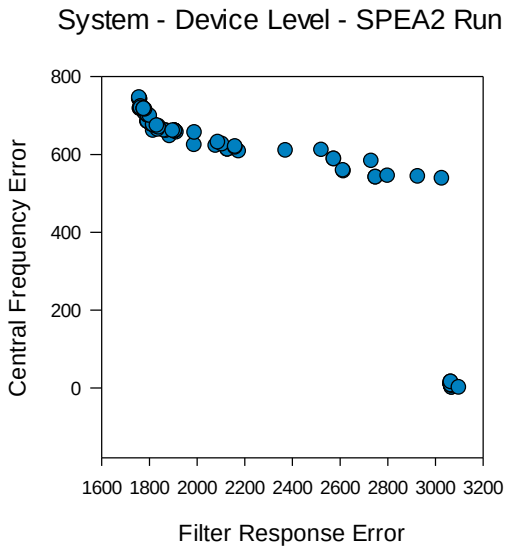
Bi-level design optimisation results

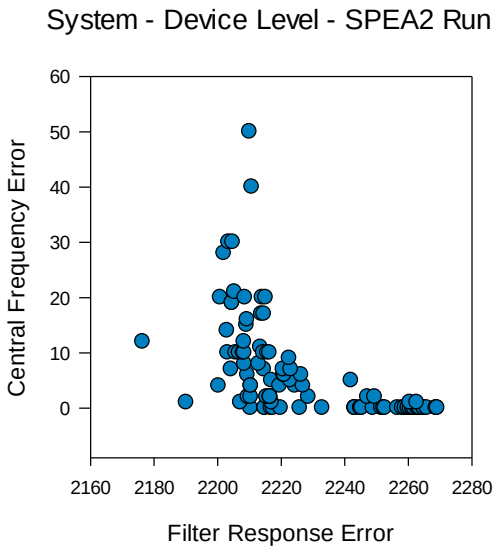
D.1 System – Device level experimental results

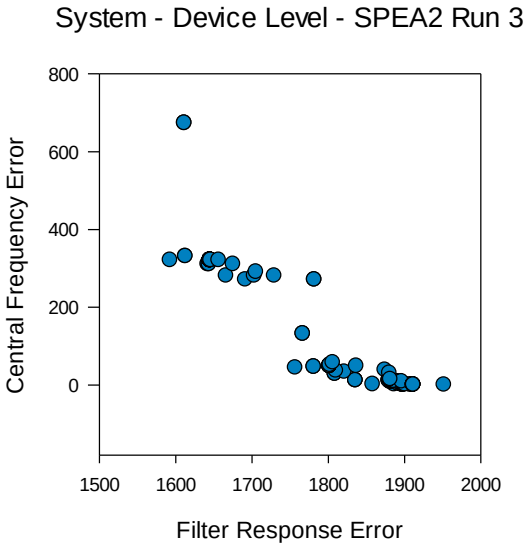
The following section holds system – device level results for the single level, multi-level evaluation, and multi-level parameterization strategies. These include final population sets and the best filter results achieved by each algorithm and strategy, along with some strategy specific analysis results.

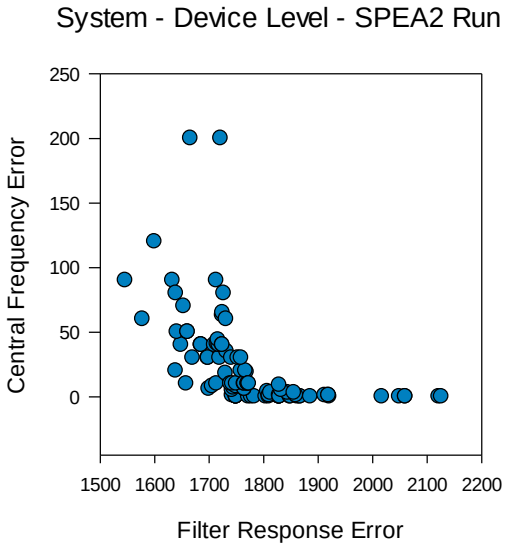


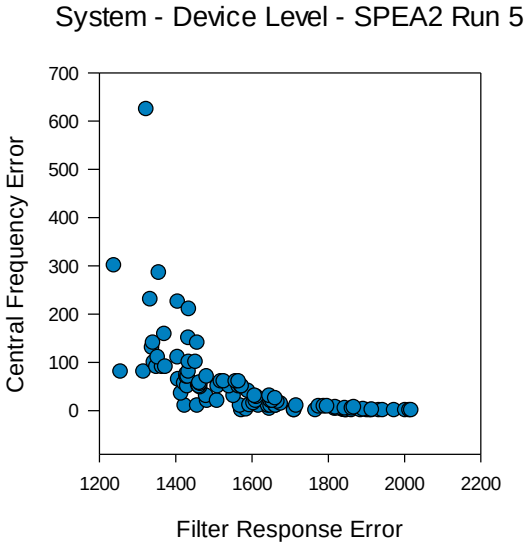
Appendix D.1 System - Device level run 1 – 5 final population sets for NSGAII

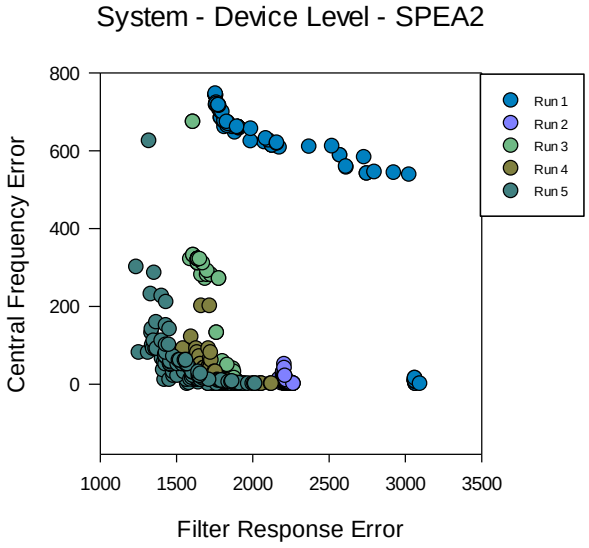




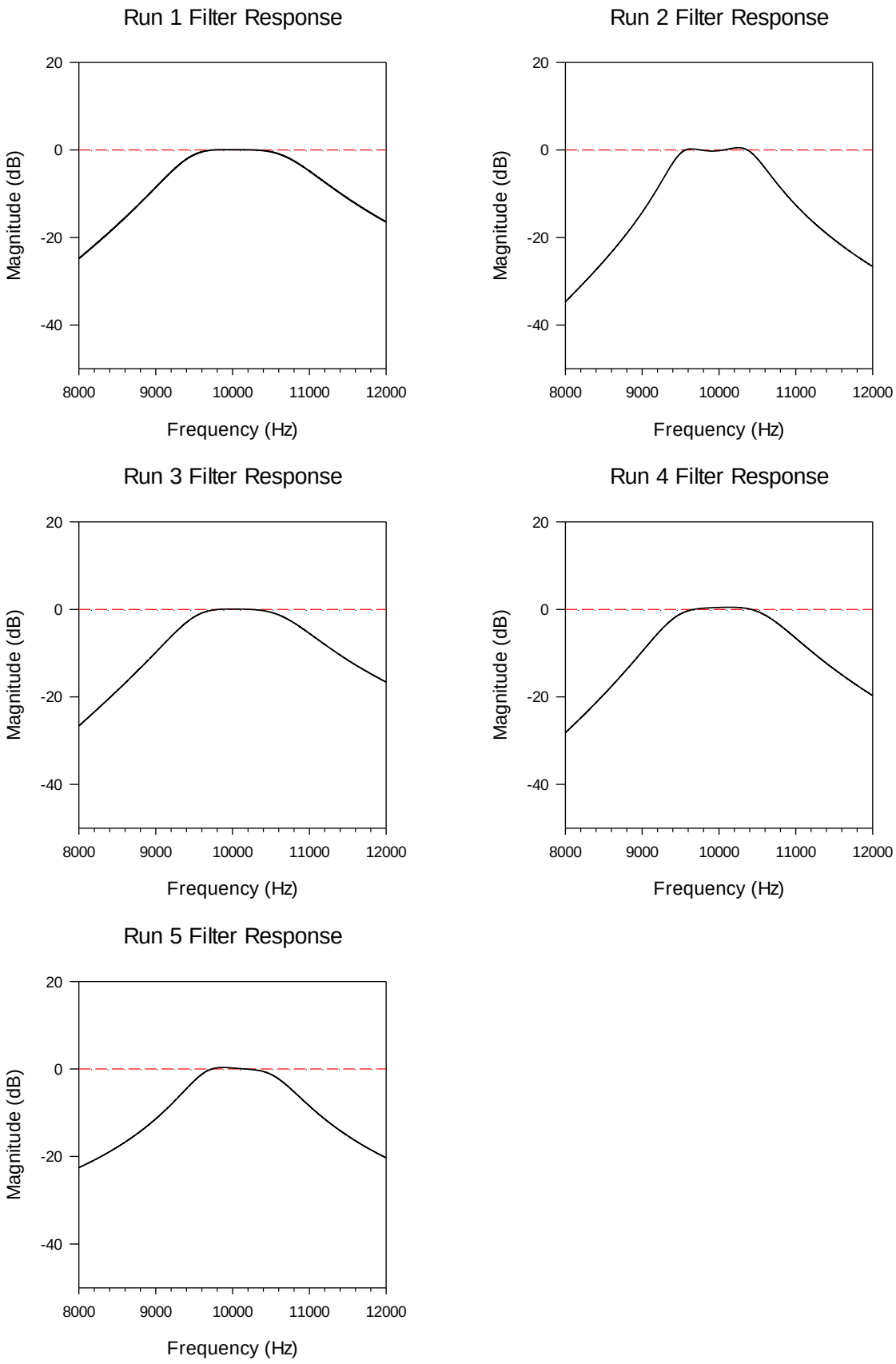




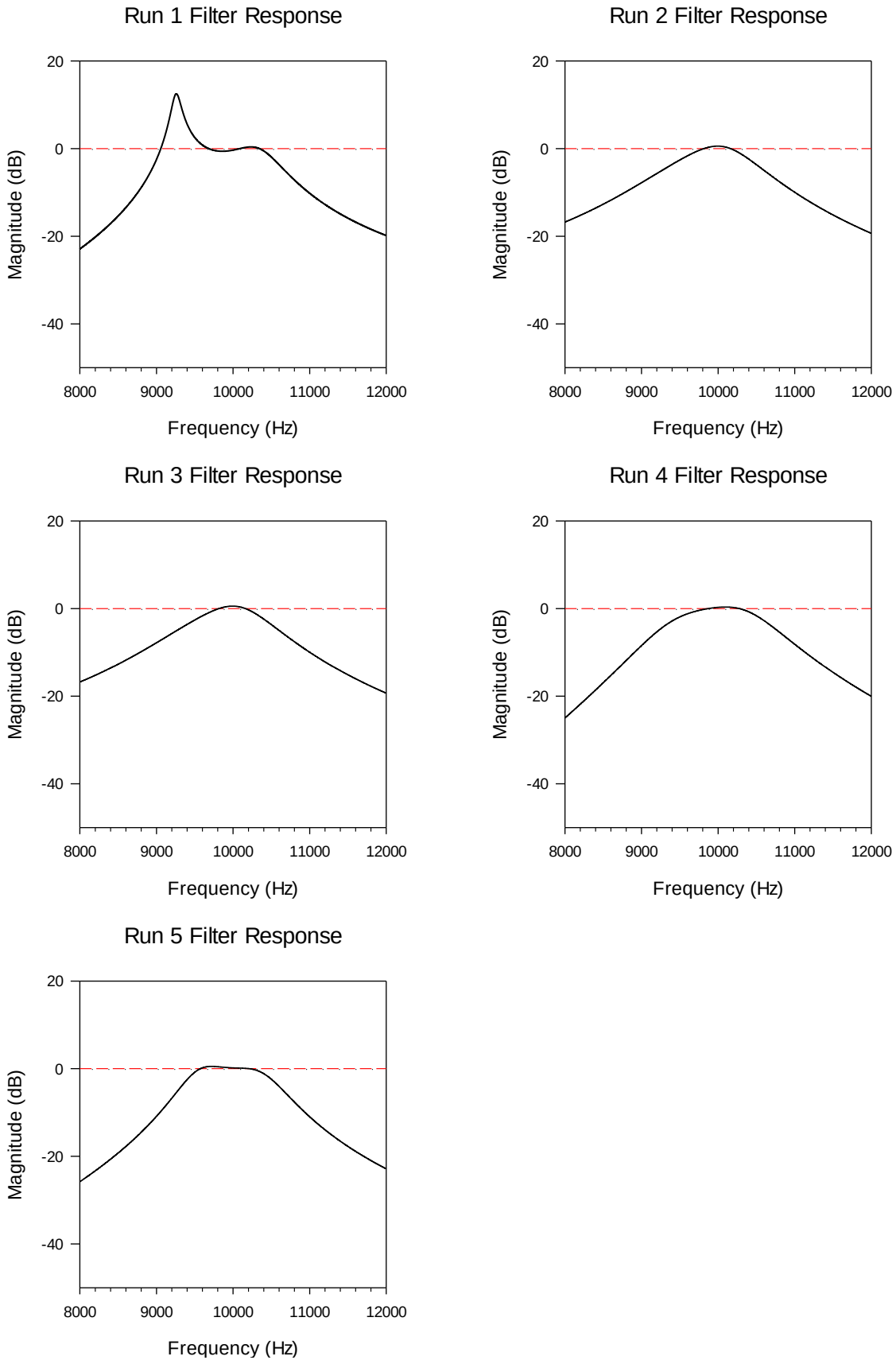




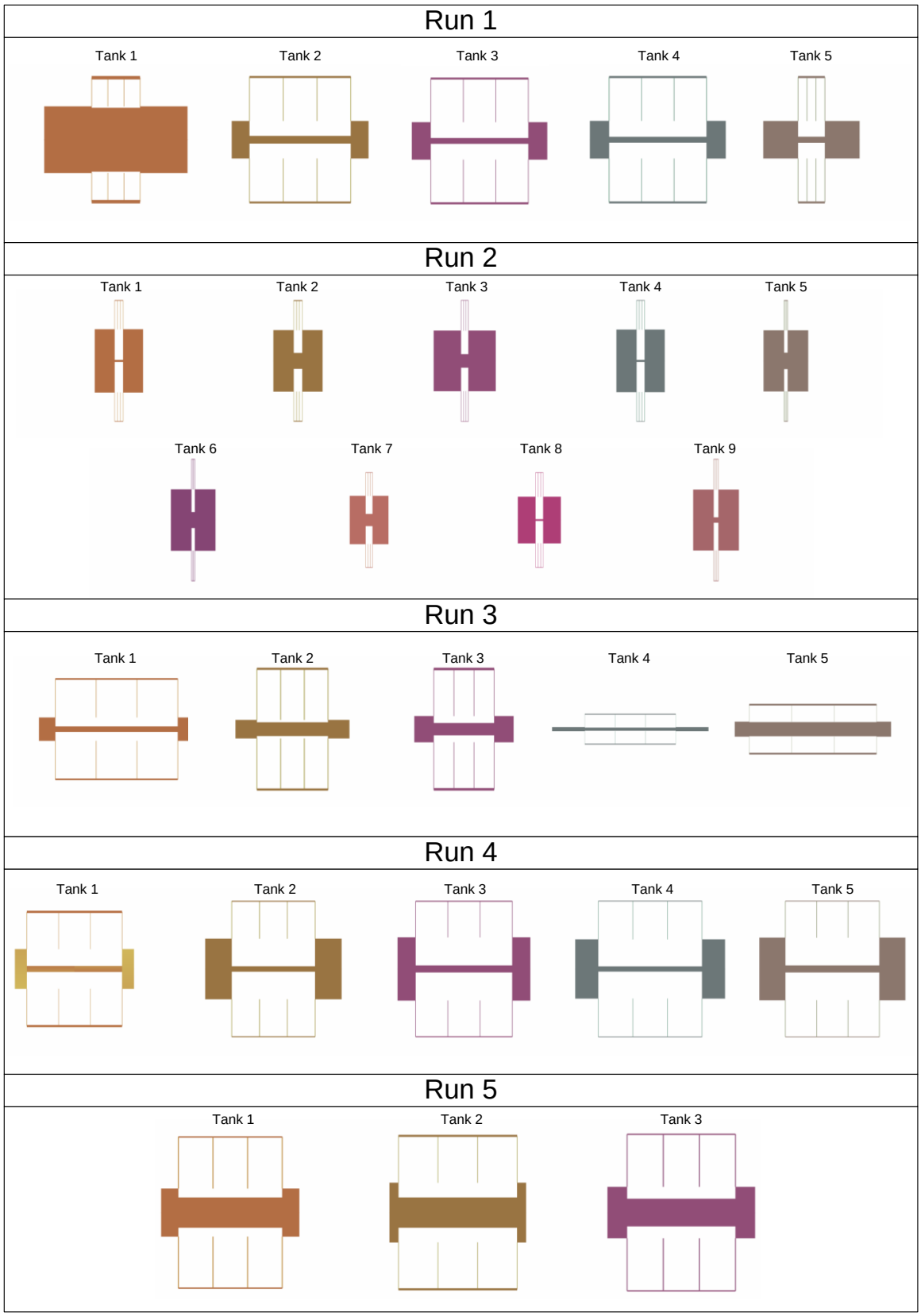
Appendix D.2 System - Device level run 1 – 5 final population sets for SPEA2



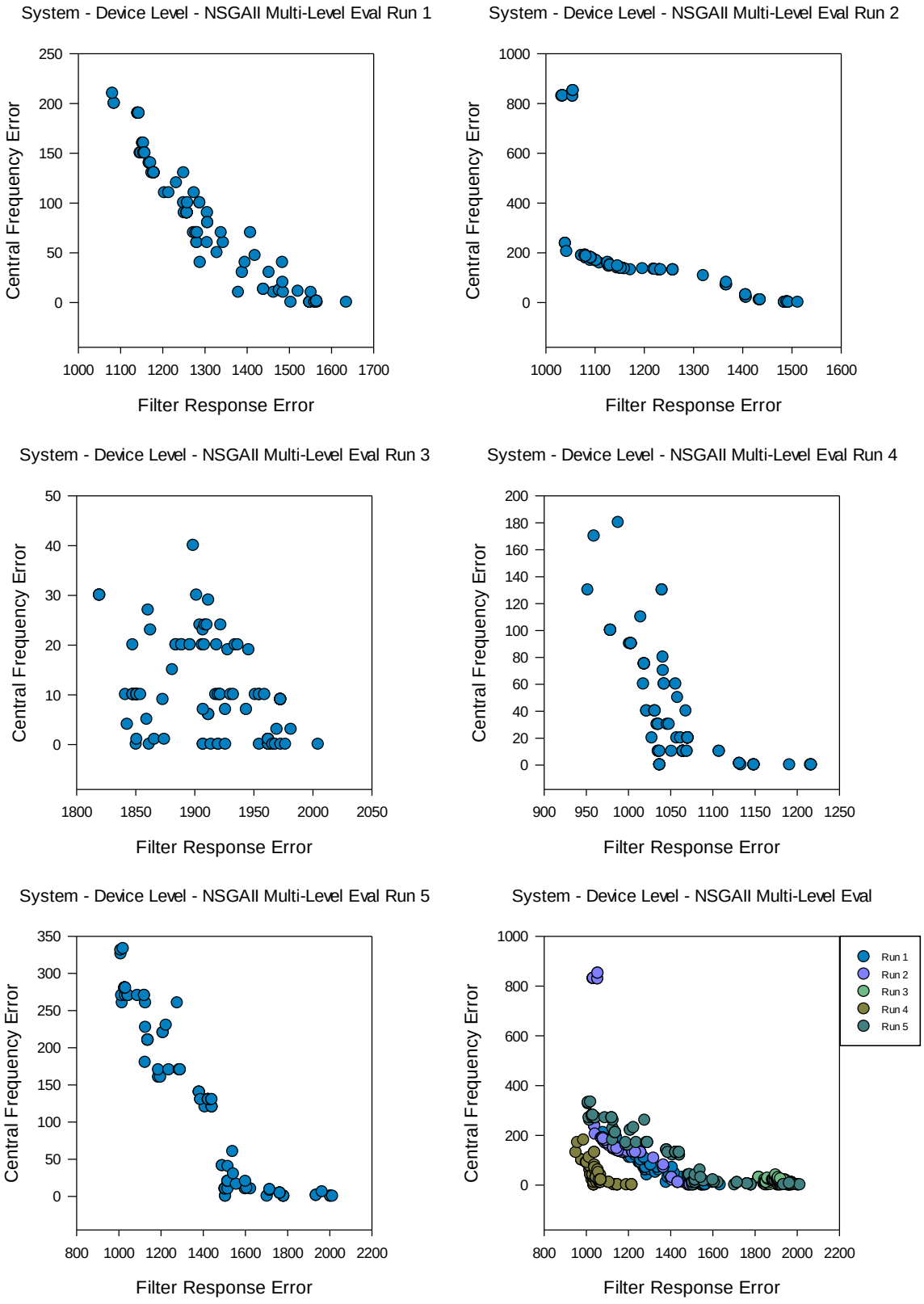
Appendix D.3 System - Device level run 1 – 5 best filter response ranked by filter frequency objective for NSGAII



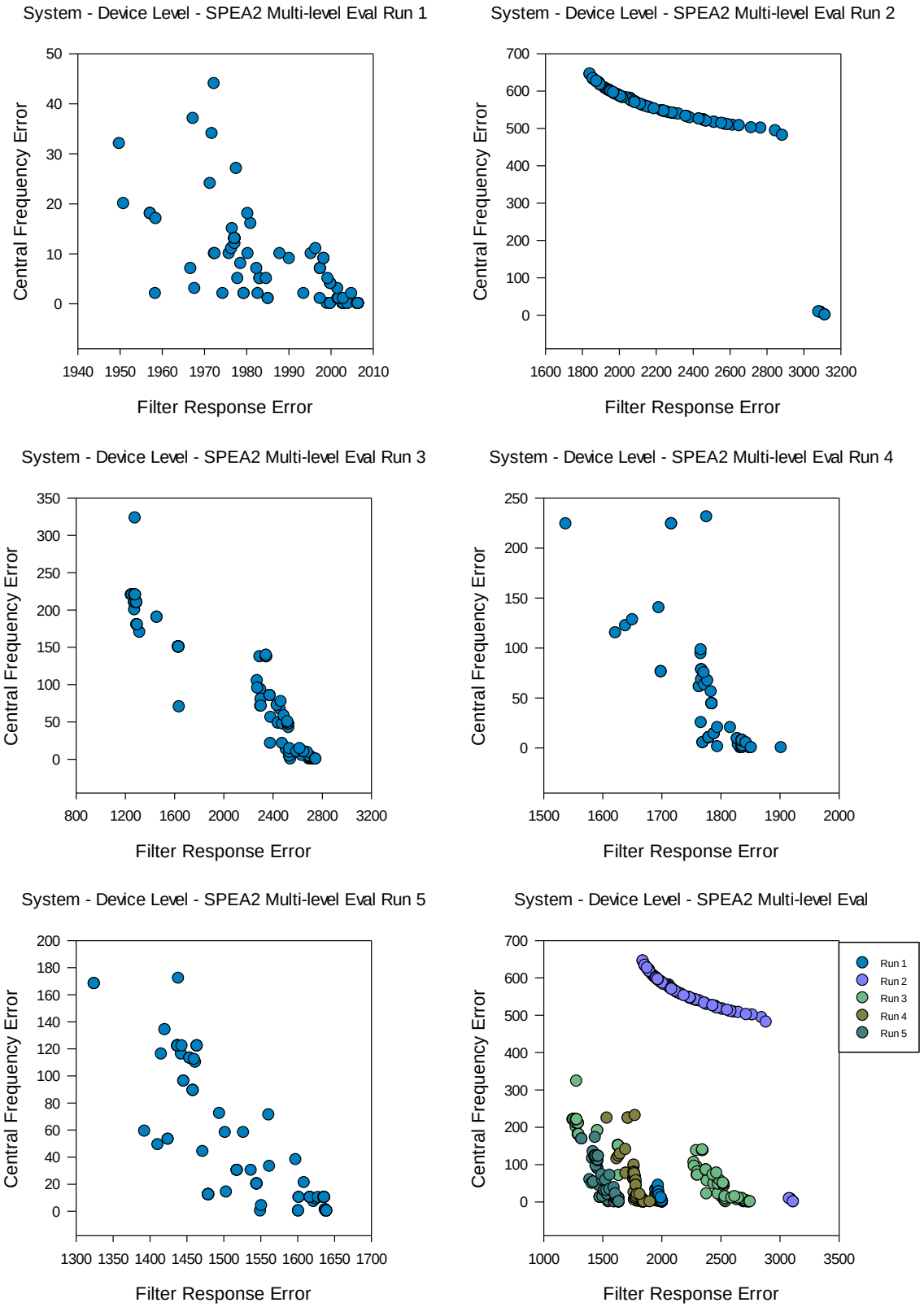
Appendix D.4 System - Device level run 1 – 5 best filter response ranked by filter frequency objective for SPEA2



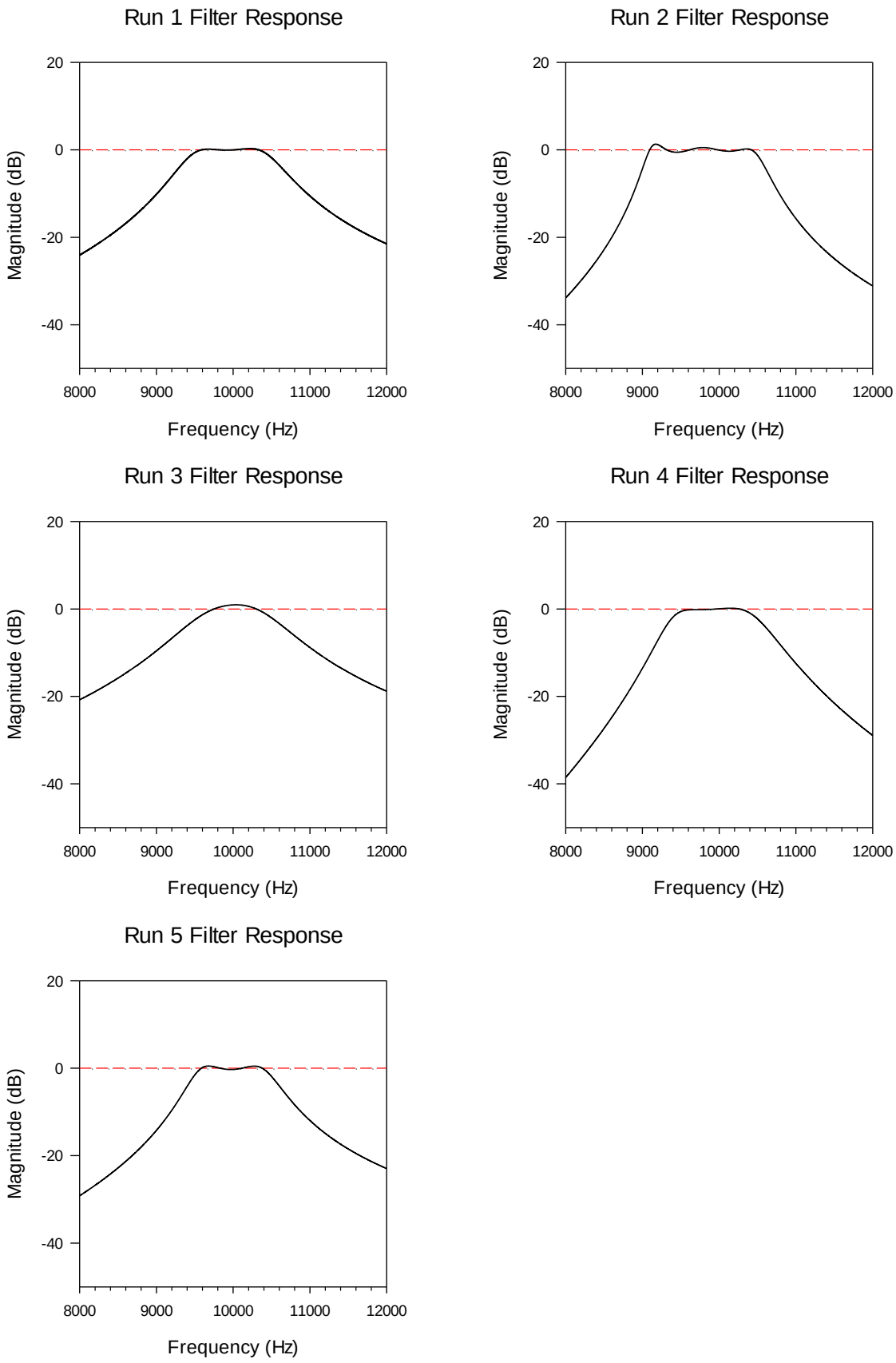
Appendix D.5 System - Device level run 1 – 5 folded flexure SPICE best result ranked by filter frequency objective for NSGAII



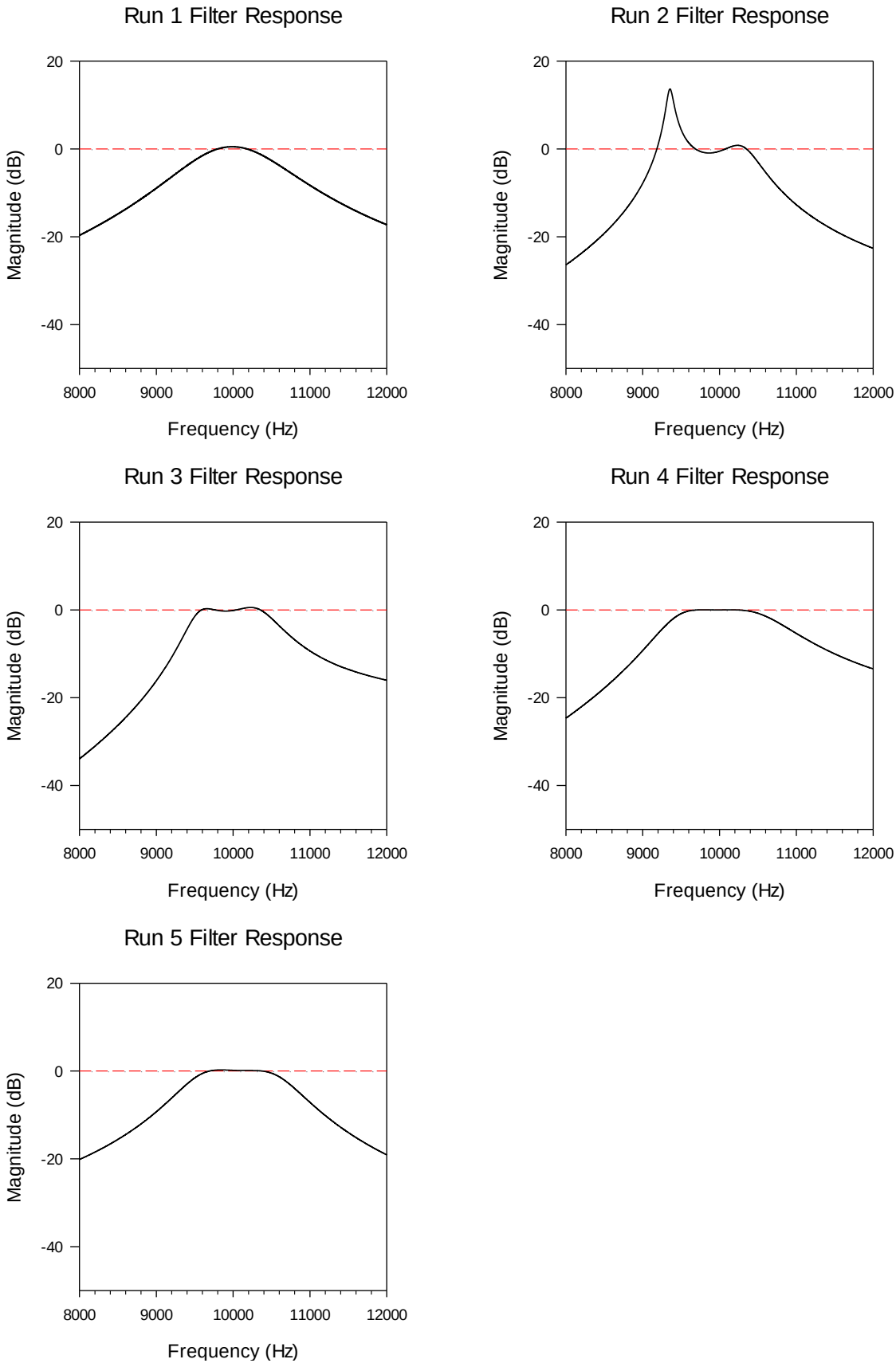
Appendix D.6 System - Device level run 1 – 5 final population sets for NSGAII multi-level evaluation



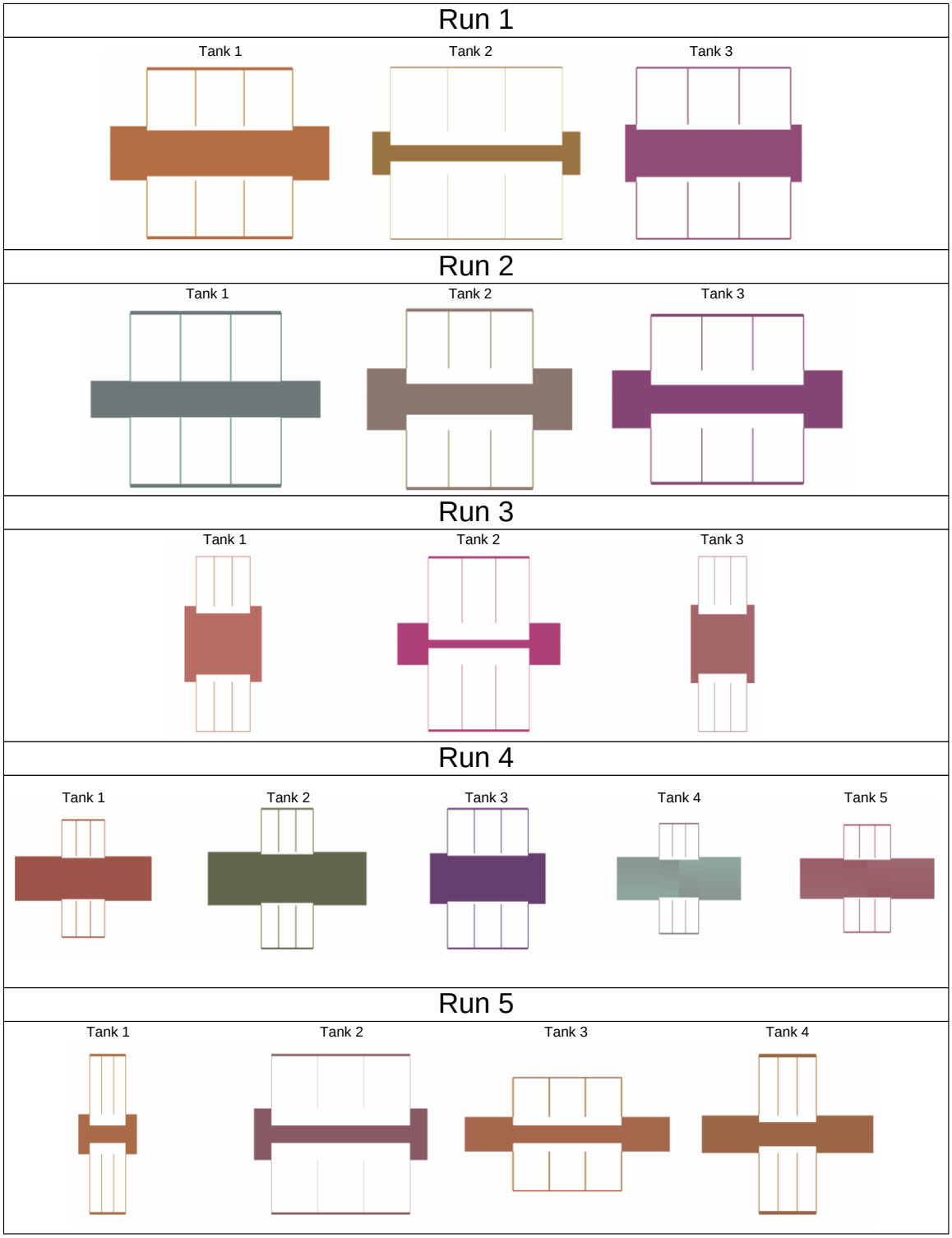
Appendix D.7 System - Device level run 1 – 5 final population sets for SPEA2 multi-level evaluation



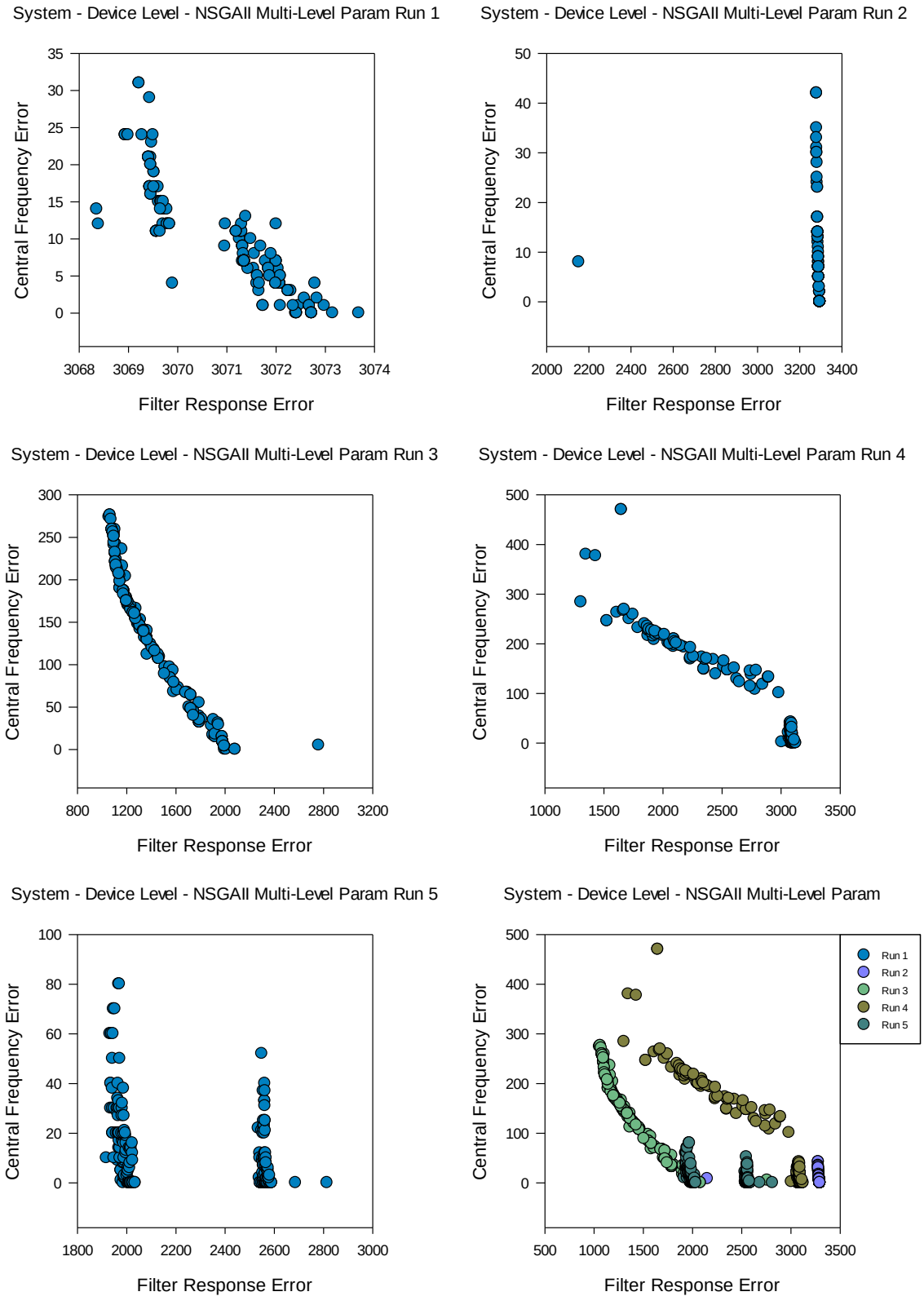
Appendix D.8 System - Device level run 1 – 5 best filter response ranked by filter frequency objective for NSGAII multi-level evaluation



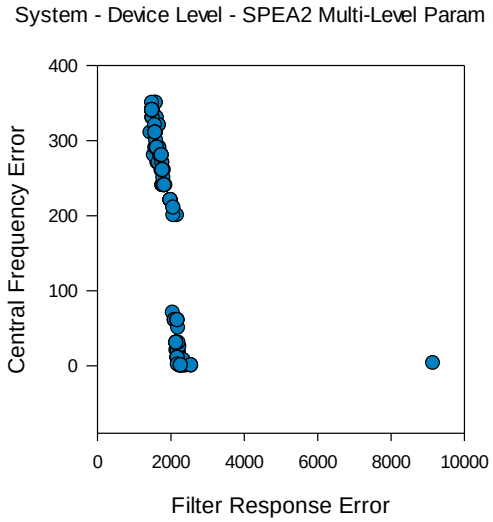
Appendix D.9 System - Device level run 1 – 5 best filter response ranked by filter frequency objective for SPEA2 multi-level evaluation

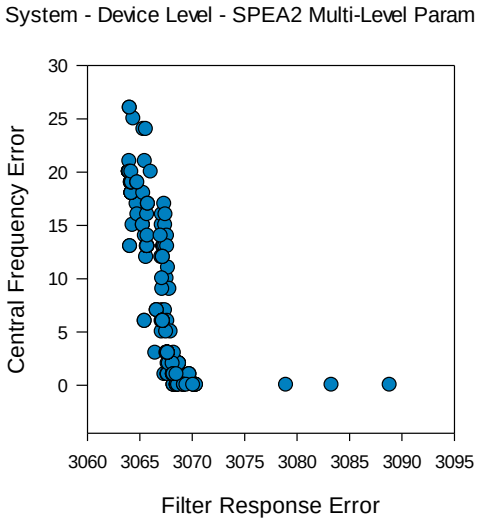


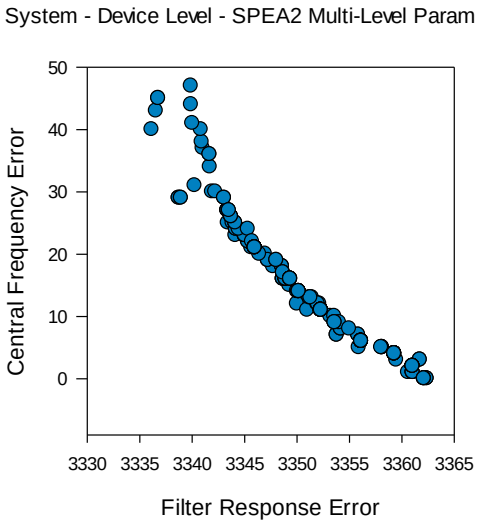
Appendix D.10 System - Device level run 1 – 5 folded flexure SPICE best result ranked by filter frequency objective for NSGAII multi-level evaluation

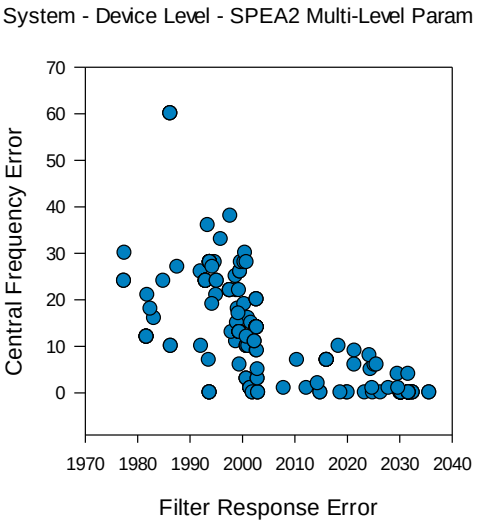


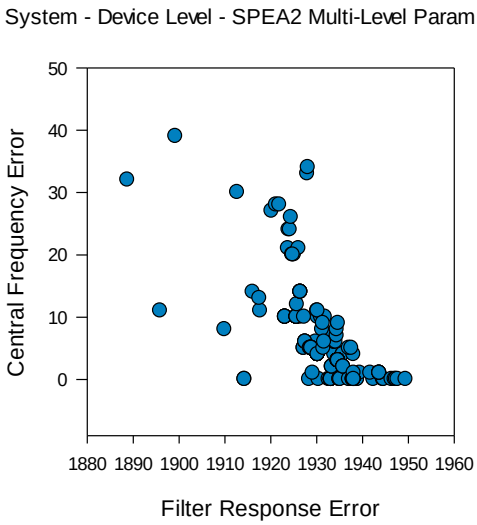
Appendix D.11 System - Device level run 1 – 5 final population sets for NSGAII multi-level parameterization

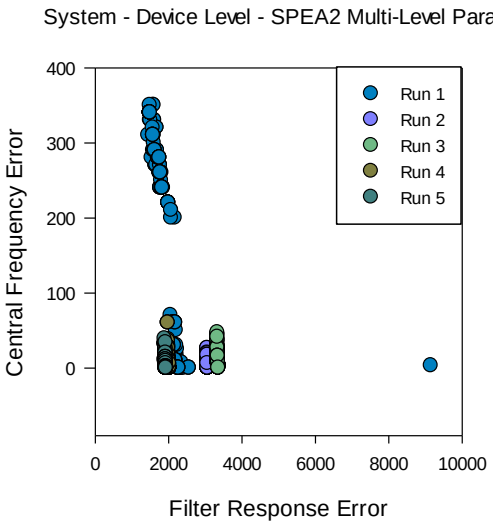




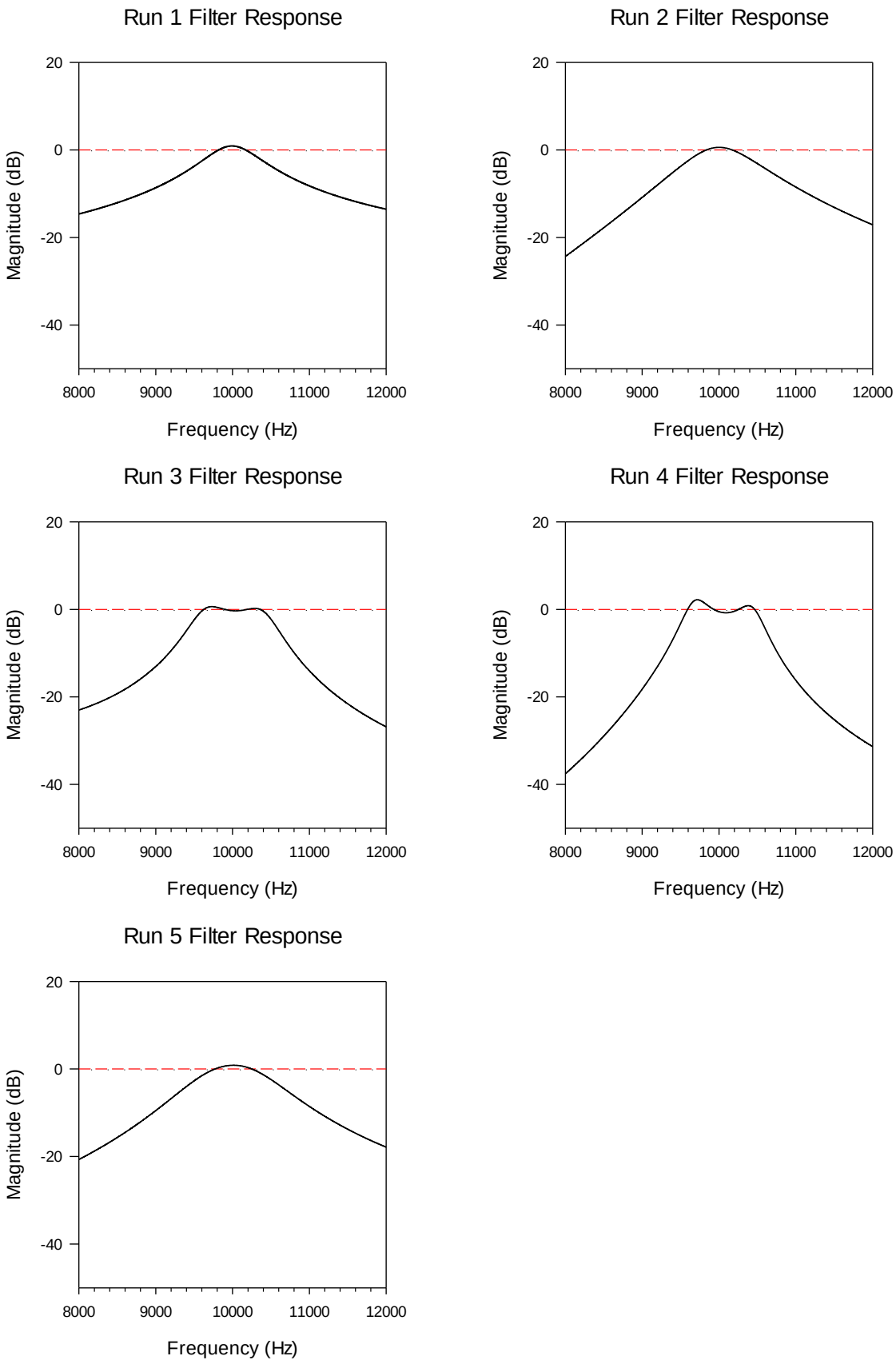




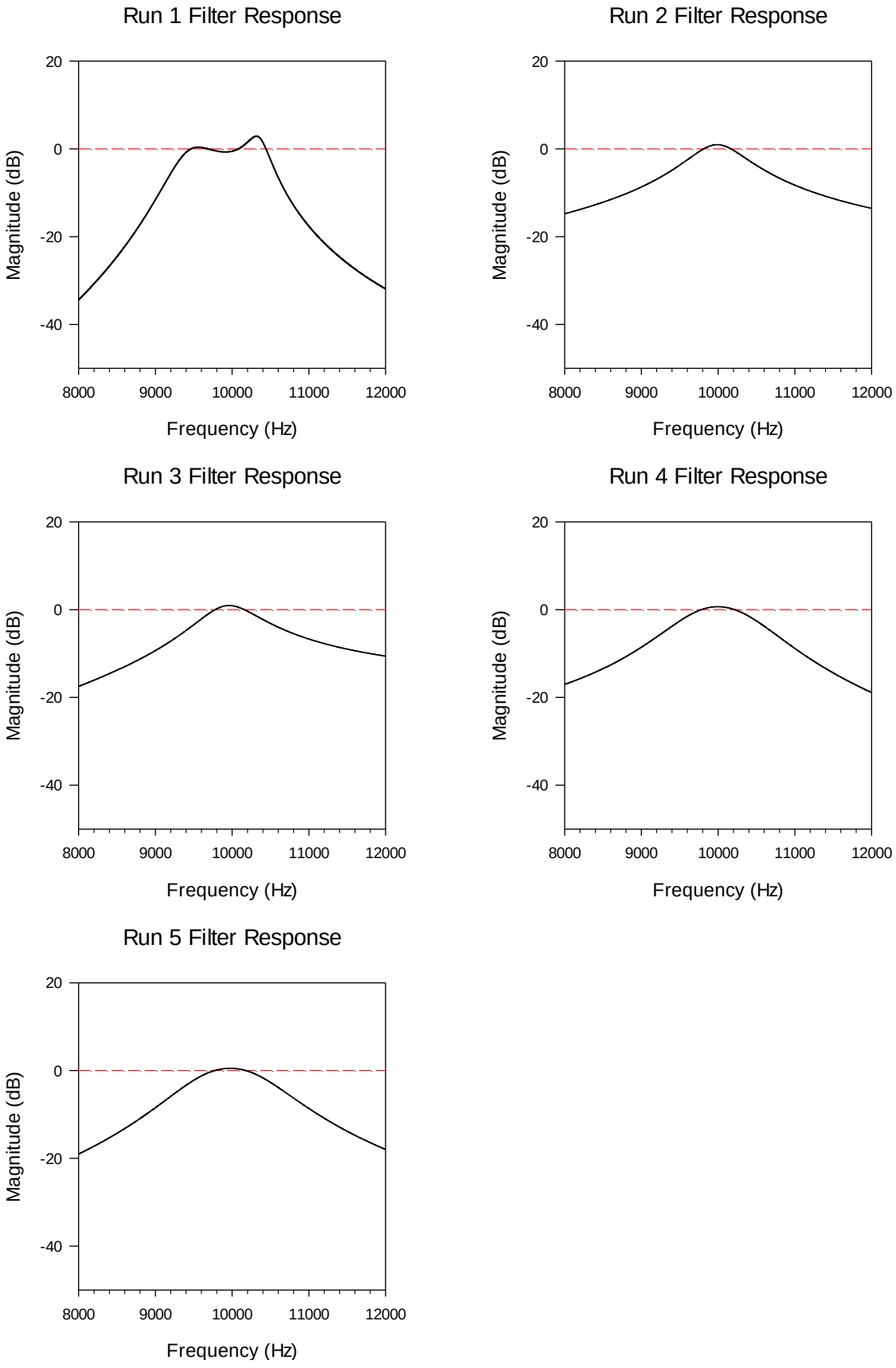




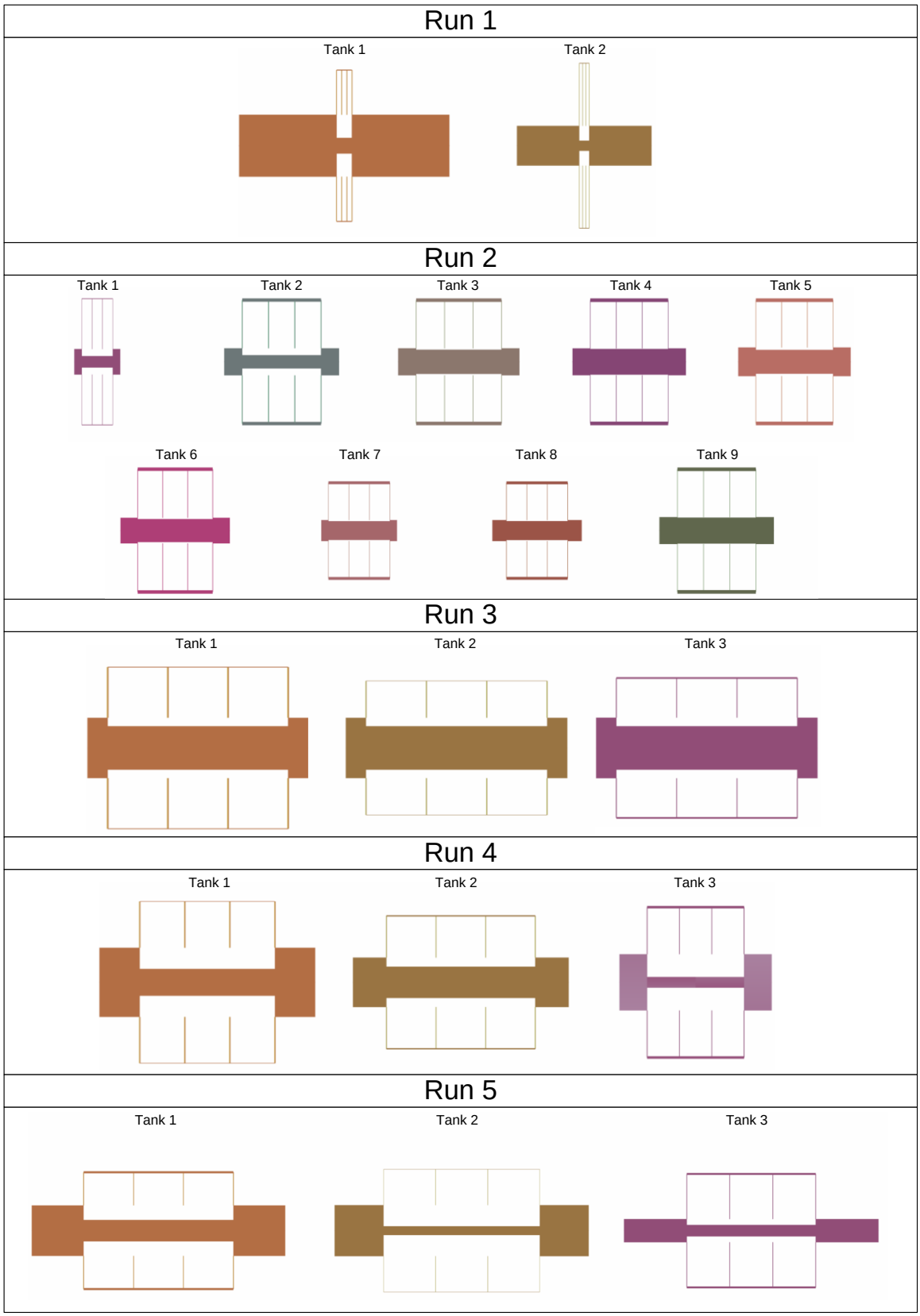
Appendix D.12 System - Device level run 1 – 5 final population sets for SPEA multi-level parameterization



Appendix D.13 System - Device level run 1 – 5 best filter response ranked by filter frequency objective for NSGAII multi-level parameterization



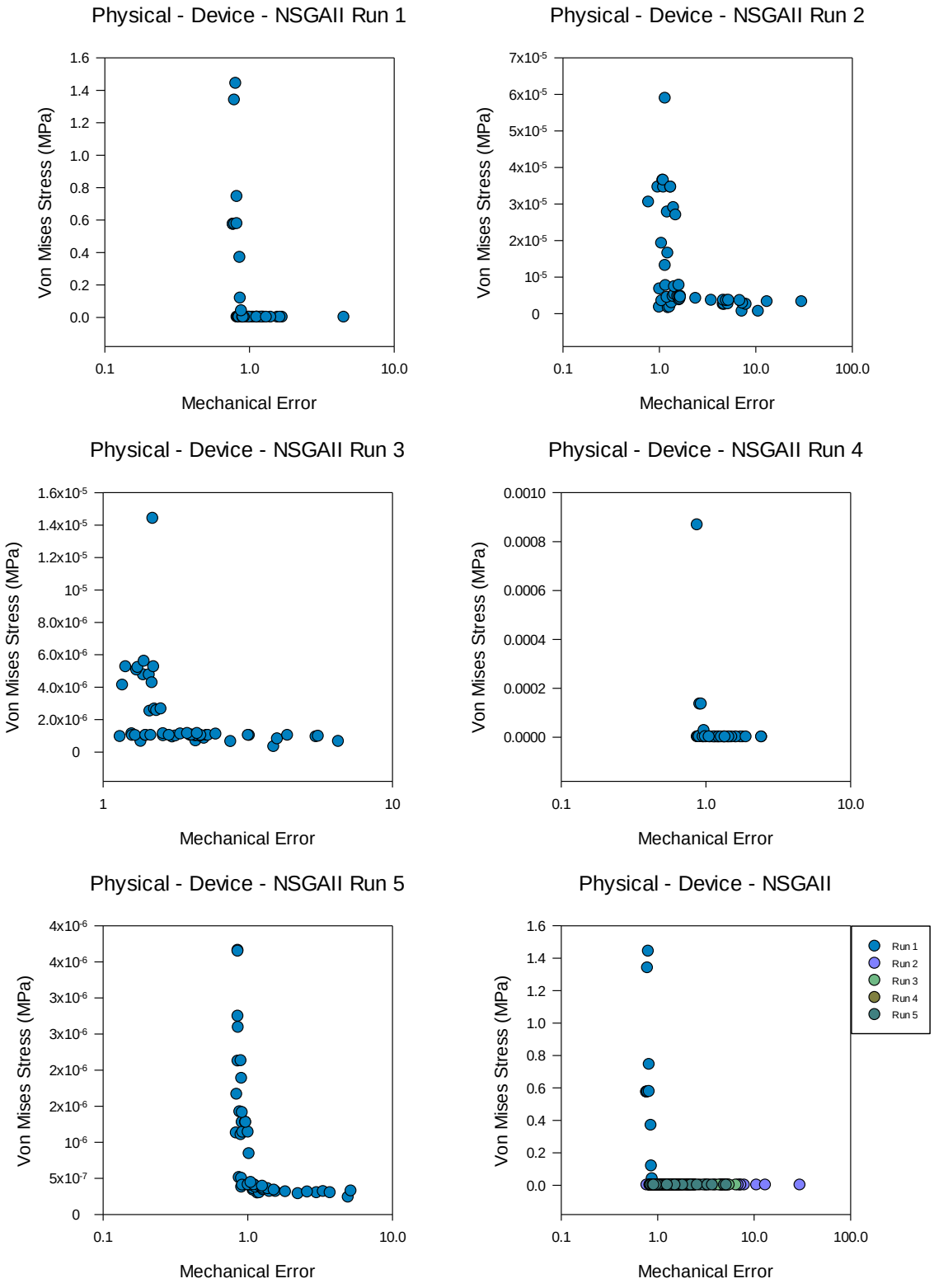
Appendix D.14 System - Device level run 1 – 5 best filter response ranked by filter frequency objective for SPEA2 multi-level parameterization



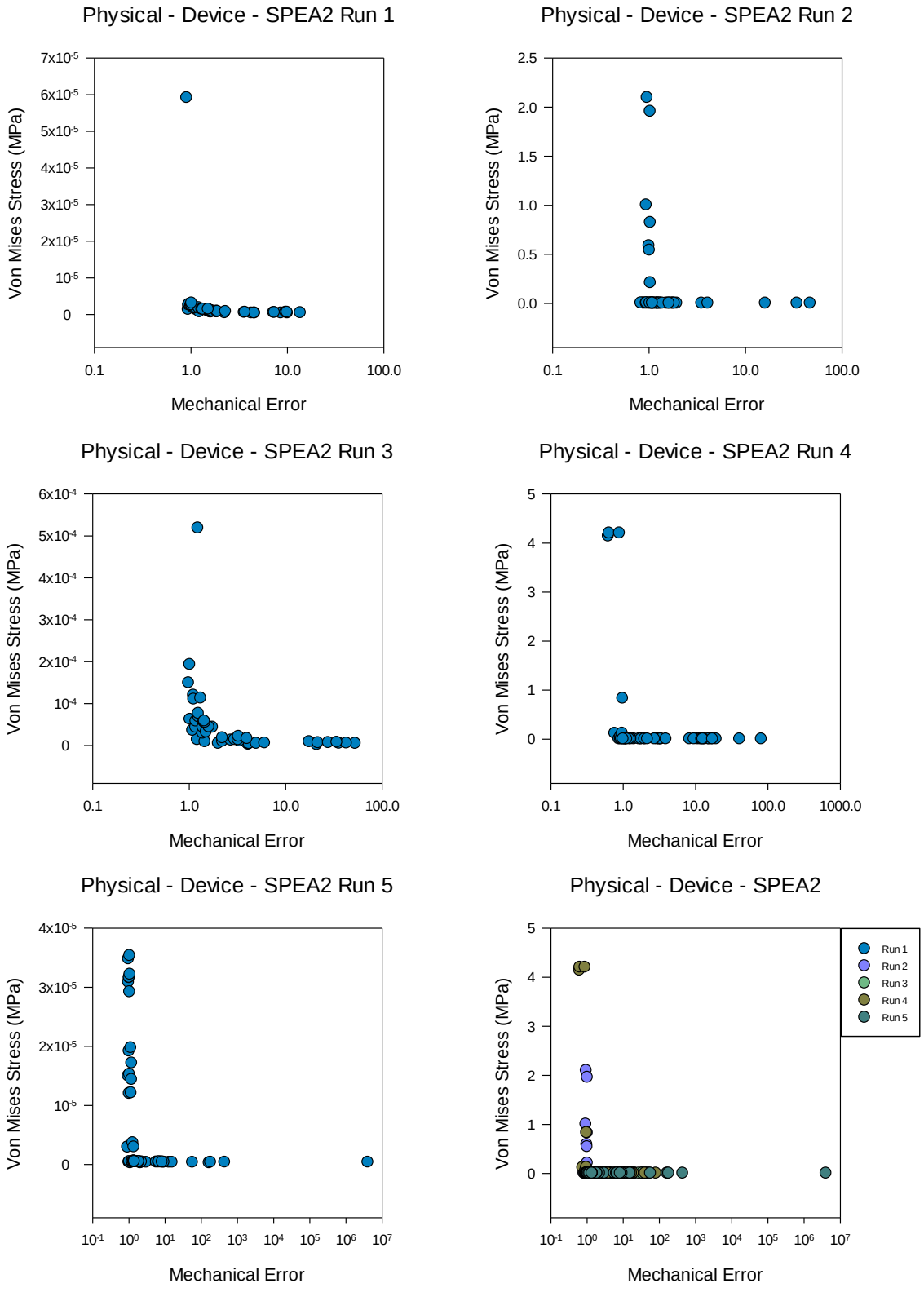
Appendix D.15 System - Device level run 1 – 5 folded flexure SPICE best result ranked by filter frequency objective for NSGAII multi-level parameterization

D.2 Physical – Device level experimental results

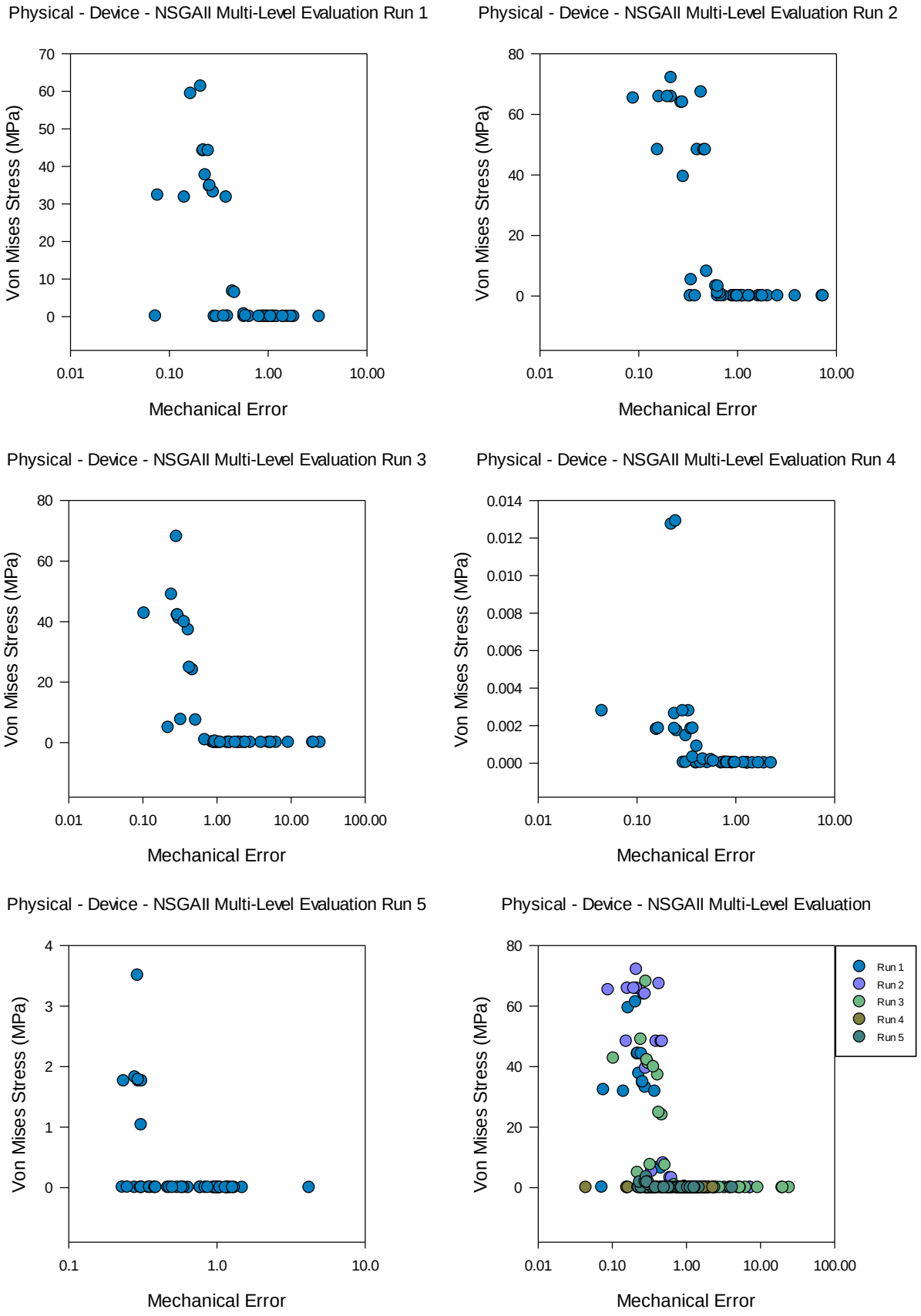
The following section holds the final population sets for the physical – device level results of the single level, multi-level evaluation, and multidisciplinary optimisation strategies.



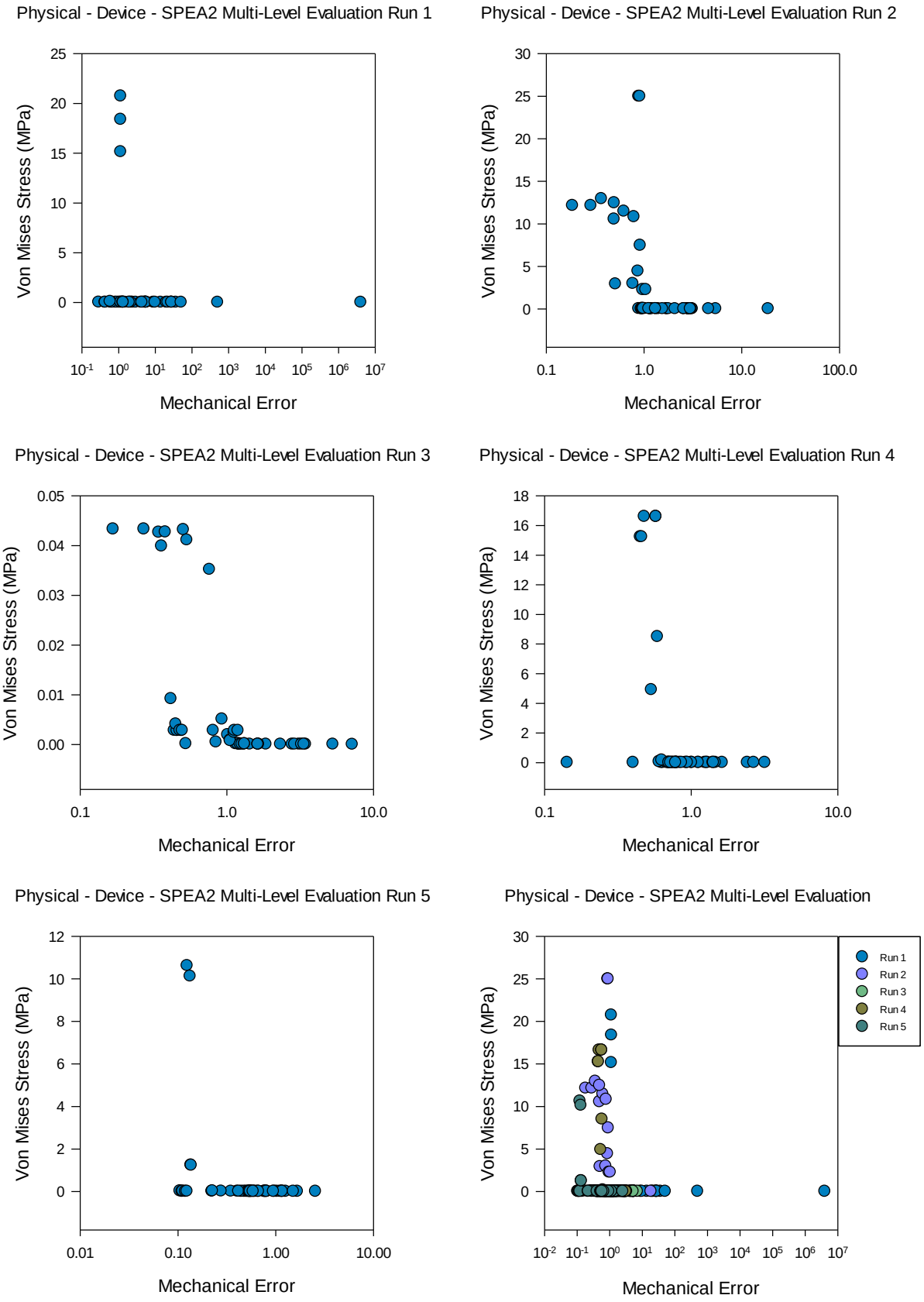
Appendix D.16 Physical – Device level run 1 – 5 final population sets for NSGAII



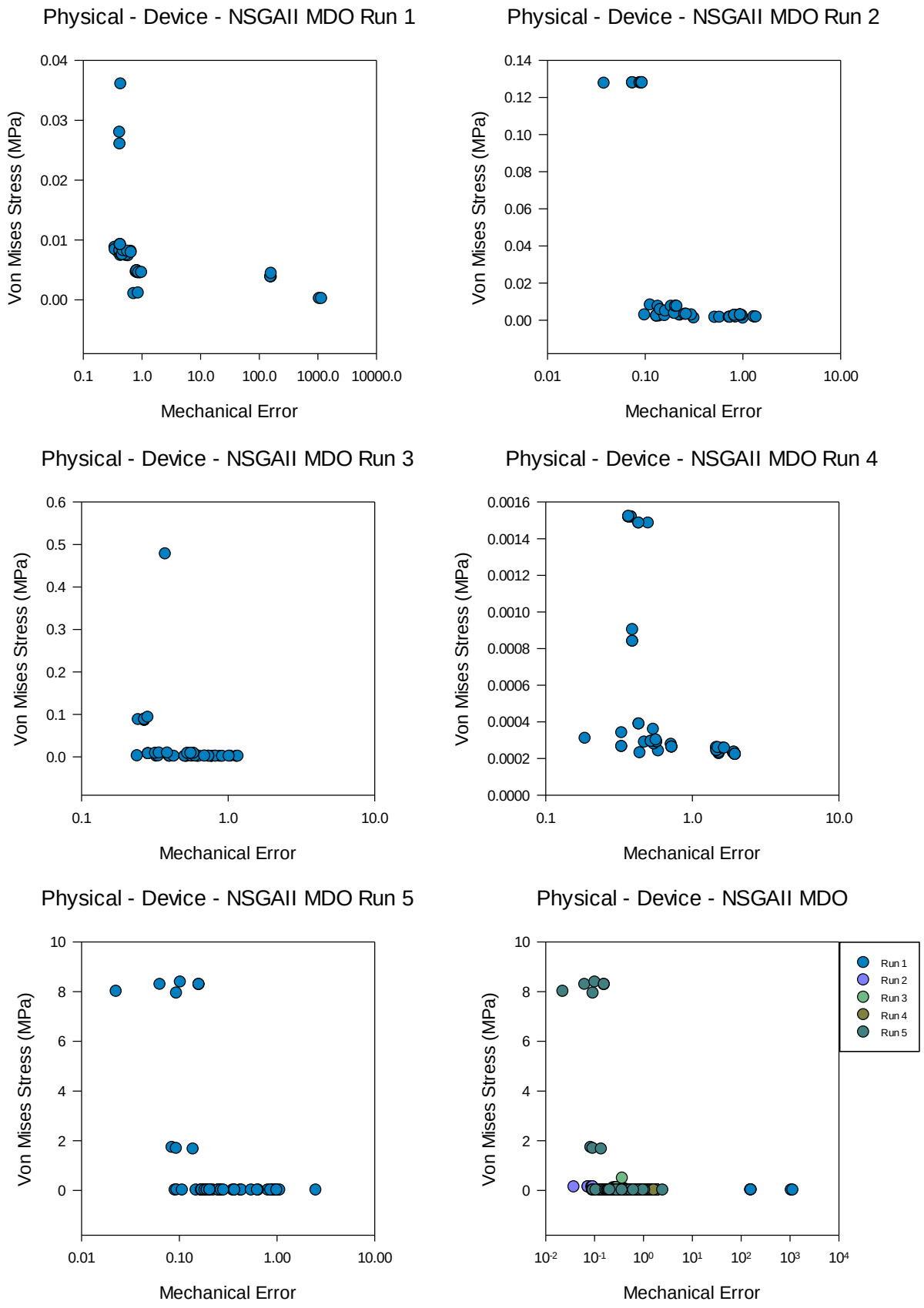
Appendix D.17 Physical – Device level run 1 – 5 final population sets for SPEA2



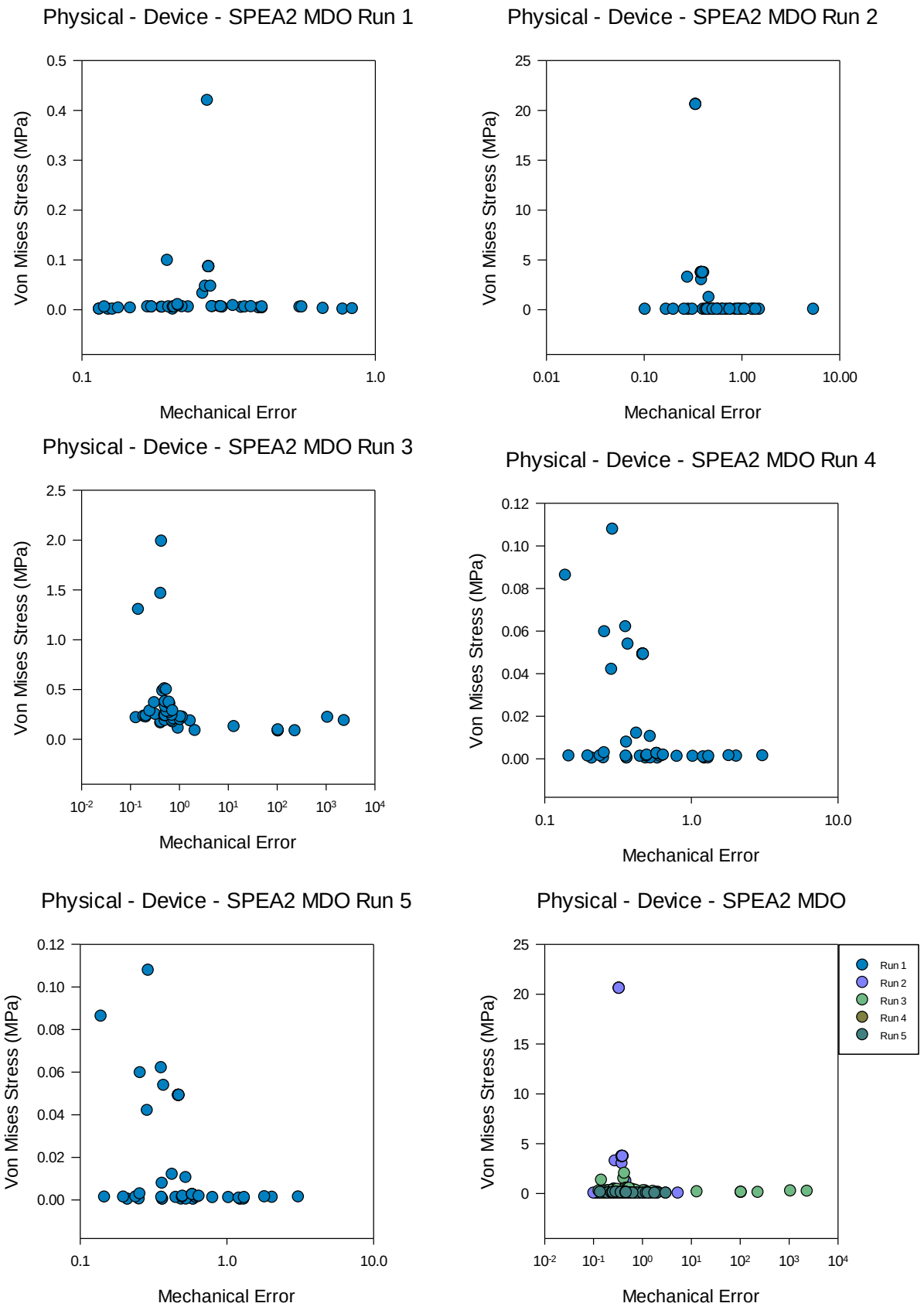
Appendix D.18 Physical – Device level run 1 – 5 final population sets for NSGAII multi-level evaluation



Appendix D.19 Physical – Device level run 1 – 5 final population sets for SPEA2 multi-level evaluation



Appendix D.20 Physical – Device level run 1 – 5 final population sets for NSGAII multidisciplinary optimisation



Appendix D.21 Physical – Device level run 1 – 5 final population sets for SPEA2 multidisciplinary optimisation

References

[1] Fujita, H., (2007) “Two Decades of MEMS-- from Surprise to Enterprise” *In Proceedings of MEMS*, Kobe, Japan, pp 21-25. January, 2007.

[2] Hsu, T. R., (2008) “MEMS and Microsystems” Second Edition. Wiley, ISBN: 978-0-470-08301-7

[3] Fedder, G. K., and Mukherjee, T., (1996) "Physical Design For Surface-Micromachined MEMS" *In Proceedings of the 5th ACM/SIGDA Physical Design Workshop*, Reston, VA USA, pp 53-60, April 15-17, 1996.

[4] Ongkodjojo, A., and Tay, F. E. H., (2002) “Global Optimization and Design for Microelectromechanical Systems Devices Based on Simulated Annealing” *Journal of Micromechanics and Microengineering,* pp 878-897.

[5] Isoda, T. and Ishida, Y. (2006) “Seperation of Cells using Fluidic MEMS Device and a Quantitative Analysis of Cell Movement” *Transactions of the Institute of Electrical Engineering of Japan*, Vol 126. No 11. pp 583-589.

[6] Neeves, K. B., and Diamond, S., (2008) “A membrane-based microfliudic device for controlling the flux of platelet agonists into flowing blood’ *Lab Chip*. 8(5). pp 701-709.

[7] Hostis, F. l., Green, N. G., Morgan, H., and Akaisi, M., (2006) “Solid state AC Electroosmosis Micro Pump on a Chip’ *International Conference on Nanoscience and Nanotechnology, ICONN*, Brisbane, Qld. pp 282-285. July, 2006.

[8] Ahn, C. H., Choi, J-W., Beaucage, G., Nevin, J. H., Lee, J-B., Puntambekar, A., and Lee, J. Y., (2004) “Disposable Smart Lab on a Chip for Point-of-Care Clinical Diagnostics” *Proceedings of the IEEE,* vol 92. No 1. pp 154-173.

[9] Esashi, M., Sugiyama, S., Ikeda, K., Wang, Y., and Myashita, H., (1998) “Vacuum-Sealed Silicon Micromachined Pressure Sensors” *Proceedings of the IEEE*, vol 86. No 8. pp 1627-1639.

[10] Zhang, Z-H., Zhang, Y-H., Liu, L-T., and Ren, T-L., (2008) “A Novel MEMS Pressure Sensor with MOSFET on Chip” *IEEE Sensors,* pp 1564-1567.

[11] Ji, C-H., Yee, Y., Choi, J., Kim, S-h., and Bu, J-U., (2004) “Electromagnetic 2 x 2 MEMS Optical Switch” *IEEE Journal of Selected Topics in Quantum Electronics*, vol 10, No 3. pp 545-550.

[12] Ponmalar, S., (2008) “Design of High Speed Optical Switches for Intelligent Optical Networks” *Proceedings of the 2008 International Conference on Computing, Communication and Networking*, pp 1-4.

[13] Potheir, A., Hitier, S., El khatib, M., Blondy, P., Orlianges, J. C., Champeaux, C., Catherinot, A., Vendier, O. and Cazaux, J. L., (2005) “MEMS DC contact micro relay on ceramic substrate for space communications switching network” *2005 European Microwave Conference*, Vol 1.

[14] Vudathu, S., and Laur, R., (2007) “A Design Methodology for the Yield Enhancement of MEMS Designs with Respect to Process Induced Variations” *Proceedings of the 57th Electronic Components and Technology Conference, ECTC '07*, Reno, NV, USA. pp 1947-1952.

[15] Fedder, G., (1999) “Structured Design of Integrated MEMS” *Micro Electro Mechanical Systems*. MEMS '99. Twelfth IEEE International Conference, Orlando, FL, USA. pp 1-8.

[16] Senturia, S. D., (2001) *Microsystem Design,* Kluwer Academic Publishers. ISBN-0-7923-7246-8.

[17] Hao, Y., and Zhang, D., (2004) “Silicon-based MEMS process and standardization” *Proceedings of the 7th International Conference on Solid-State and Integrated Circuits Technology*. Vol 3. pp 1835-1838

[18] IntelliSuite (1995) Available at http://www.intellisensesoftware.com, [accessed 11/2010]

[19] MEMSPro (1997) ‘softMEMS’, available at http://www.softmems.com/ [accessed 11/2010]

[20] Senturia, S. D., Harris, R. M., Johnson, B. P., Kim, S., Nabors, K., Shulman, M. A., and White, J. K., (1992) “A computer-aided design system for microelectromechanical systems (MEMCAD)”, *IEEE Journal of MicroElectroMechanical Systems*, March, Vol. 1, pp 3–13.

[21] Antonsson, E. K (1996) “Structured Design Methods for MEMS” *Final Report National Science Foundation Workshop*. Nov 12-15, Californian Institute of Technology.

[22] Vudathu, S., and Laur, R., (2007) “A design methodology for the yield enhancement of MEMS designs with respect to process induced variations” *Proceedings of the 57th Electronic Components and Technology Conference, (2007), ECTC '07*, pp 1947‑1952, Reno, NV, USA.

[23] Schneider, P., Schneider, A., and Schwarz, P., (2002) “A modular approach for simulation-based optimization of MEMS” *Microelectronics Journal*, Vol 33, pp 29-38.

[24] Leman, O., Dumas, N., Mailly, F., Latorre, L., and Nouet, P., (2008) “An approach to integrate MEMS into high-level system design flows*” In Circuits and Systems and TAISA Conference*, pp 273-276.

[25] Fedder, G. K., (2000) “Top-Down Design of MEMS” *Technical Proceedings of the 2000 International Conference on Modeling and Simulation of Microsystems*, pp 7-10.

[26] McCorquodale, M. S., Gebara, F. H., Kraver, K. L., Marsman, E. D., Senger, R. M., and Brown, R. B., (2003) “A Top-Down Microsystems Design Methodology and Associated Challenges” *In Proceedings of the Design, Automation and Test in Europe Conference and Exhibition.*

[27] Pingyu, J., Xiangtong, Y., and Lui, Y., (2007) “A web “top-down” design tool for creating micro-components” *Journal of Manufacturing Technology Management*, Vol 18, pp 90-105.

[28] Mukherjee, T., (2003) “MEMS Design and Verification” *In ITC International Test Conference*, pp 681-690.

[29] Haronain, D., (1995) "Maximizing microelectromechanical sensor and actuator sensitivity by optimizing geometry", *Sensors and Actuators A*, 50, pp. 223-236

[30] Iyer, S., Mukherjee, T., and Fedder, G., "Automated Optimal Synthesis of Mi-croresonators", *Solid-State Sensors and Actuators*, Chicago, IL, p12-19. (1997)

[31] Kamalian, R., Zhou, N., and Agogino, A. M., (2002) “A Comparison of MEMS Synthesis Techniques”, *Proceedings of the 1st Pacific Rim Workshop on Transducers and Micro/Nano Technologies*, July 22-24, Xiamen, China. pp 239-242.

[32] Zhou, N., Agogino, A. M., and Pister, K. S., (2002) “Automated Design Synthesis for Micro-Electro-Mechanical Systems (MEMS)”, *Proceedings of the ASME Design Automation Conference*, ASME CD ROM, Sept. 29-Oct. 2, Montreal, Canada.

[33] Kamalian, R. H., Takagi, H., and Agogino, A. M., (2004) "Optimized Design of MEMS by Evolutionary Multi-objective Optimization with Interactive Evolutionary Computation" *Proceedings of GECCO 2004* (Genetic and Evolutionary Computation Conference; June 26-30, Seattle, Washington), CD ROM.

[34] Karakasis, M., and Giannakoglou, K., (2003) “Inexact information aided, Low-cost, distributed genetic algorithms for aerodynamic shape optimisation” *International Journal for Numerical Methods in Fliuds*, Vol 43, (10-11), pp 1149-1166.

[35] El-Beltagy, M. A., and Keane, A., (1999) “Topographical mapping assisted Evolutionary Search for Multilevel Optimization” *In Proceedings of the 1999 Conference on Evolutionary Computation.*

[36] Abou El Majd, B., Duvigneau, R., and Desideri, J.-A., (2006) “Aerodynamic shape optimisation using a full and adaptive multilevel algorithm” *In ERCOFTAC Conference Design Optimization : Methods and Applications*, Canary Island, Spain, April.

[37] Duvigneau, R., (2007) “A Multi-Level Particule Swarm Optimization strategy for aerodynamic shape optimization” *Evolutionary Methods for Design, Optimization and Control*. In CIMNE, Barcelona, Spain.

[38] Martinelli, M., and Beux, F. (2006) “Optimum shape design through multilevel gradient-based method using Bezier parameterisation” *In 4th ICCFD Conference*, Ghent, Belgium, July.

[39] Desideri, J-A., and Janka, A., (2003) “Hierarchical parameterization for multilevel evolutionary shape optimization with application to aerodynamics” *In International Congress on Evolutionary Methods for Design, Optimization and Control with application to Industrial Problems*, EUROGEN.

[40] Tosserams, S., Etman, L. F. P., and Rooda, J. E. (2010) “A micro-accelerometer MDO benchmark problem” Structural Multidisciplinary Optimisation, Vol 41, pp 255-275

[41] Li, H. and Antonsson, E.K. (1999) “Mask-layout synthesis through an evolutionary algorithm” *MSM’99, Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators*, April, San Juan, Puerto Rico.

[42] Zhang, Y., Kamalian, R., Agogino, A. M., and Séquin, C. H., (2006) “Design Synthesis of Microelectromechanical Systems Using Genetic Algorithms with Component-Based Genotype Representation” *In Proc. of GECCO 2006 (Genetic and Evolutionary Computation Conference)*. Seattle, July 8-12, 2006. ISBN 1-59593 187-2. Vol. 1, pp 731-738

[43] Kamalian, R (2004) “Evolutionary synthesis of MEMS devices”, PhD thesis, University of California, Berkeley, California, United States of America.

[44] Poles, S., (2003) "MOGA-II An Improved Multi-Objective Genetic Algorithm" *Technical report 2003-006*, Esteco, Trieste.

[45] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., (2000) “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II". *In Proceedings of the 6-th International Conference Parallel Problem Solving from Nature (PPSN-VI),* pp 849-858.

[46] Kampolis, I., and Giannakoglou, K., (2008) “A Multilevel approach to single- and multiobjective aerodynamic optimization” *Computer Methods in Applied Mechanics and Engineering*, 197. (33-40), pp 2963-2975

[47] Achiche, S., Fan, Z. and Bolognini, F., (2007) “Review of Automated Design and Optimization of MEMS”, *IEEE International Symposium on Industrial Electronics*, June pp 2150-2155

[48] Benkhelifa, E., Farnsworth, M., Bandi, G., Tiwari, A., Zhu, M., and Ramsden, J., (2010) "Design and Optimisation of Microelectromechanical Systems: A Review of the State-of-the-Art" *International Journal of Design Engineering, Special Issue "Evolutionary Computing for Engineering Design"* Vol 3, No 1. pp 41-76

[49] Mukherjee, T., Fedder, G., Ramaswamy, D., and White, J., (2000) "Emerging Simulation Approaches For Micromachined Devices", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*. pp 1572-1589

[50] Farnsworth, M., Benkhelifa, E., Tiwari, A. and Zhu, M., (2010) "A Novel Approach to Multi-level Evolutionary Design Optimization of a MEMS Device" Evolvable Systems: From Biology to Hardware, LNCS, vol 6274, pp 322-334

[51] Matlab-Simulink, Mathworks (online) http://www.mathworks.com/ [accessed 11/2010]

[52] Fan, Z., Seo, K., Hu, J., Rosenberg, R. C., and Goodman, E. D., (2003) 'System Level Synthesis of MEMS via Genetic Programming and Bond Graphs' Genetic and Evolutionary Computation GECCO. (Springer Berlin / Heidelberg) pp 205-217

[53] Fan, Z., Wang, J., Achiche, S., Goodman E., and Rosenberg R. (2008) 'Structured Synthesis of MEMS using evolutionary approaches' Applied Soft Computing 8, pp 579-589

[54] Van Kuijk, J., Schropfer, G. and DaSilva, M. (2005) 'Design Automation for MEM/MST', *Design, Automation and Test in Europe Conference & Exhibition*, Munich, Germany.

[55] Nagel, L. W, and Pederson, D. O., (1973) *SPICE (Simulation Program with Integrated Circuit Emphasis)*, Memorandum No. ERL-M382, University of California, Berkeley, Apr.

[56] Verliog-AMS, Accellera (online) http://www.eda.org/verilog-ams/ [accessed 12/2008]

[57] ELDO, Mentor Graphics (online) http://www.mentor.com/ [accessed 12/2008]

[58] Cadence, (online) http://www.cadence.com/ [accessed 12/2008]

[59] Broenink, F., (1999) "Introduction to Physical Systems Modeling with Bond Graphs, SiE Whitebook on Simulation Methodologies", http://www.rt.el.utwente.nl/bnk/papers/BondGraphsV2.pdf.

[60] Getreu, I. (1989) 'Behavioral Modelling of Analog Blocks using the SABER Simulator', *Proceedings on Microwave Circuits and Systems*, pp.977-980, Aug

[61] Chwirka, S. (2000) 'Using the powerful SABER simulator for simulation, modeling and analysis of power systems, circuits, and devices', *IEEE Xplore*.

[62] Swart, N.R., Bart, S. F., Zaman, M.H., Mariappan, M., Gilbert, J. R. and Murphy, D. (1998) 'AutoMM: Automatic Generation of Dynamic Macromodels for MEMS Devices', *IEEE Proceedings of the Eleventh Annual International Workshop on Microelectromechanicalsystems*. pp.178-183

[63] Lee, K.W. and Wise, K. D, (1982) 'A simulation program for solid state pressure sensors', *IEEE Trans. Electron. Devices*, ED-29, pp.34-41.

[64] Bin, T. Y. and Huang, R. S, (1987) 'CAPPS: thin diaphragm capacitive pressure sensor simulator' *Sensors and Actuators*, 11 pp 1-22.

[65] Zhu, M., Kirby, P. and Lim, M. Y. (2004) 'Langrange's Formalism for Modeling of a Triaxial Microaccelerometer with Piezoelectric Thin-Film Sensing' *IEEE Sensors Journal*, vol 4, No 4. Aug. pp.455-463

[66] Zhou, N., Clark, J.V., and Pister, K.S.J., (1998) 'Nodal Analysis for MEMS Design Using Sugar v0.5', *Technical Proceedings of the 1998 International Conference on Modelling and Simulation of Microsystems*, pp.308-313

[67] Nagel. L. W., (1970) "Computer analysis of nonlinear circuits excluding radiation", Ph.D dissertation, UC Berkeley. California, 1970

[68] Vandemeer, J., Kranz, M. and Fedder, G, (1997) 'Nodal Simulation of Suspended MEMS Multiple Degrees of Freedom', *MEMS ASME*: DSC-Vol. 62, pp.113-118.

[69] Vandemeer, J., Kranz, M. and Fedder, G, (1998) 'Hierarchical Representation and Simulation of Micromachined Inertial Sensors', *Modeling and Simulation of Microsystems*, Santa Clara, CA, April

[70] MAST hardware language (online) http://www.synopsys.com/ [accessed 12/2008]

[71] ABAQUS Alliances: Coventor (online) "http://www.hks.com/alliances/alliances_coventor.html" [accessed 12/2008]

[72] Clark, J.V., Zhou, N. and Pister, K.S. J, (1998) 'MEMS Simulation Using SUGAR v0.5', *Solid-State Sensor and Actuator Workshop*, pp.191-196.

[73] Coventor (online) "http://www.coventor.com/mems/architect/index.html" [accessed 12/2008]

[74] Bechtold, T., Rudnyi, E. B. and Korvink, J. G. (2003) 'Automatic Generation of Compact Electro-thermal Models for Semiconductor Devices', *IEICE Transactions on Electronics*, vol. 86, pp.459-65,

[75] Chen, Y. and White, J. (2000) 'A Quadratic Method for Nonlinear Model Order Reduction', *Proceedings of Modelling and Simulation of Microsystems*, pp.477-80

[76] Ying, Y.J. and Yu, C.C. (2004) 'Extraction of heat-transfer macromodels for MEMS devices', *Journal of Micromechanics and Microengineering,* vol 14, No 4, pp. 587-596.

[77] Wang, F. and White, J, (1998) 'Automatic Model Order Reduction of a Microdevices Using the Arnoldi Approach', *ASME International Mechanical Engineering Congress and Exposition, Proceedings of Microelectromechanical Systems (MEMS)*, Anaheim, CA, Nov. pp.527-30

[78] Silveira, L.M., Kamon, M., Elfadel, I. and White, J. (1999) 'A Coordinate-Transformed Arnoldi Algorithm for Generating Guaranteed Stable Reduce-Order Models of RLC Circuits', *Computer Methods in Applied Mechanics and Engineering*, vol 169, no 3-4 (19 ref), pp.377-389

[79] Rewienski, M. and White, J. (2001) 'A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuit and Micromachined Devices', *Proceedings of Computer-Aided Design*, pp.252-257

[80] Hung, E., Yang, Y-J. and Senturia, S.D. (1997) 'Low-Order Models for Fast Dynamical Simulation of MEMS Microstructures', *Proceedings of Transducers'97*, pp.1101-1104

[81] Antoulas, A. C., (2000) *Approximation of linear Dynamical Systems,* Wiley Encyclopaedia of Electrical and Electronics Engineering (Ed: J.G. Webster), 11, pp.684-687

[82] Puers, B., Petersen, E. and Sansen, W, (1989) 'CAD Tools in Mechanical Sensor Design,' *Sensors and Actuators A*, v.A17, pp.423-429.

[83] Crary, S. and Zhang, Y, (1990) 'CAEMEMS: An Integrated Computer-Aided Engineering Workbench for Micro-Electrical-Mechanical Systems', *Proceedings of the IEEE Workshop on MicroElectroMechanical Systems*, Napa, CA, February, pp.113-114.

[84] Crary, S. and Zhang, Y, (1991) 'Software Tools for Designers of Sensor and Actuator CAE Systems', *Solid-State Sensors and Actuators*, 23-27: pp.498-501.

[85] Zhang, Y., Crary, S.B. and Wise, K.D, (1990) 'Pressure Sensor Design and Simulation Using the CAEMEMS-D Module,' *Technical Digest of the IEEE Solid-State Sensor and Actuator Workshop*, *June 1990,* Hilton Head Is., SC.

[86] Folkmer, B., Offereins, H.-L. , Sandmaier, H., Lang, W., Groth, P. and Pressmar R, (1992) 'A Simulation Tool for Mechanical Sensor Design,' *Sensors and Actuators A*, v.A32, pp.521-524.

[87] SolidWorks (online) http://www.solidworks.com [accessed 01/2009]

[88] Funk, J.M., Korvink, J.G., Buhler, B., Bachtold, M., and Baltes, H, (1997) 'SOLIDIS: A Tool for Microactuator Simulation in 3-D', *Journal of Microelectromechanical Systems*, 6(1), pp.207-212.

[89] ANSYS (online) "http://www.ansys.com" [accessed 12/2008]

[90] ALGOR, MEMS Simulation, (online) "http://www.algor.com/products/applications/mems/default.asp" [accessed 12/2008]

[91] COMSOL (online) http://www.comsol.com/ [accessed 12/2008]

[92] Koppelman, G, K, (1989) 'Oyster: A three-Dimensional Structural Simulator for Microelectromechanical Design', *Sensors and Actuators*, 20: pp.179-185.

[93] Wesley, M.A., Lozano-Perez, T., Lieberman, L.I., Lavin, M.A. and Grossman, D.D, (1980) 'A Geometric Modeling System for Automated Mechanical Assembly,' *IEM Journal* of *Research and Development,* vol. 24, no. **1,** January

[94] Calis, M. and Desmulliez, M, (2005) 'Haptic sensing technologies for a novel design methodology in micro/nanotechnology', *Nanotechnol. Perceptions* 1 pp.89-97.

[95] Buser, R.A. and de Rooij, N.F, (1991) 'ASEP: A CAD program for Silicon Anisotropic Etching (Micromechanical Structure)', *Sensors and Actuators*, pp.71-78.

[96] Buser, R.A., Crary, S.B., and Juma, O.S, (1992) 'Integration of the anisotropic-silicon-etching program ASEP within the CAEMEMS CAD/CAE framework' In *MEMS '92*, pp.133–138 Institute of Electrical Engineers

[97] Gilbert, J. R., Osterberg, P.M., Harris, R.M., Ouma, D.O., Cai, X., Pfajfer, A., White, J. and Senturia, S, D, (1993) 'Implementation of a MEMCAD System for Electrostatic and Mechanical Analysis of Complex Structures from Mask Descriptions', *IEEE Micro Electro Mechanical Systems Workshop*, pp.207-212.

[98] Osterberg P.M. and Senturia, S.D, (1995) '`MEMBUILDER': An Automated 3D Solid Model Construction Program for Microelectromechanical Structures,' *Technical Digest of the 8th Int. Conf. on Solid-State Sensors and Actuators (Transducers `95)*, *June 1995*, Stockholm Sweden, v.2, pp.21-24

[99] Hubbard, T. J., and Antonsson, E., K. (1996) 'Design of MEMS via Efficient Simulation of Fabrication', *Proceedings of the 1996 ASME Design Engineering Conference*, pp.18-22.

[100] He, Y., Harris, R., Napadenski, G. and Maseeh, F, (1996) 'IntelliCAD: A Virtual Prototype Manufacturing Software System for MEMS', *Proceedings of IEEE Micro Electro Mechanical Systems Workshop*.

[101] Zhang., C., Jiang, Z., Lu., D. and Ren, T., (2006) '3D MEMS Design Method via SolidWorks' *Proceedings of the 1st IEEE International Conference on Nano/Micro Engineered and Molecular Systems*. pp.747 - 751

[102] Brenner, M.P., Lang, J.H., Li, J., Qiu, J. and Slocum, A. H, (2002) 'Optimum Design of MEMS Switch', *Modeling and Simulation of Microsystems*

[103] Haronain, D, (1995) 'Maximizing microelectromechanical sensor and actuator sensitivity by optimizing geometry', *Sensors and Actuators A, 50*, pp. 223-236.

[104] Iyer, S., Mukherjee, T. and Fedder, G, (1997) 'Automated Optimal Synthesis of Microresonators', *Solid-State Sensors and Actuators*, Chicago, IL, pp.12-19.

[105] Mukherjee, T., Iyer, S.V. and Fedder, G, (1998) 'Optimization-based Synthesis of Microresonators', *Sensors and Actuators A: Physical*, October, pp.118-127.

[106] Krassow, H., Zabala, M., Gotz, A. and Cane, C, (1998) 'MEMS Design Optimization using coupled FEM and Electrical Circuit Simulation' *Technical Proceedings of the 1998 International Conference on Modelling and Simulation of Microsystems*, pp.329-333

[107] Ye, W., Mukherjee, **S.** and MacDonald, N.C, (1998) 'Optimal Shape Design of an Electrostatic Comb Drive in Microelectromechanical Systems', *Journal. Microelectromechanical System,* March vol. 7, pp.16-26

[108] Mukherjee, T., Zhou, Y. and Fedder, G, (1999) 'Automated Optimal Synthesis of Microaccelerometers', *Technical Digest, IEEE International MEMS Conference, Orlando, FL, Jan. 17-21, 1999*, pp326-331.

[109] Gibson, D., Purdy, C., Hare, A. and Beyette, F. Jr, (1999) 'Design Automation of MEMS Systems Using Behavioral Modeling'. *Proceedings of the IEEE Great Lakes Symposium on VLSI*, pp.266-269

[110] Parkinson, M.B., Jensen, B.D. and Roach, G.M, (2000) 'Optimization-Based Design of a Fully-compliant Bistable Micromechanism'. ASME - *Proceedings of DETC and Computers and Information in Engineering Conference.* Baltimore, Maryland.

[111] Sedivec, P.J, (2000) *Robust Optimization: Design in MEMS*. University of California. Berkeley M.Sc

[112] Han J.S. and Kwak, B. M, (2001) 'Robust optimal design of a vibratory microgyroscope considering fabrication errors'. *Journal of Micromechanics and Microengineering.* 11, pp.662-71

[113] Peano, F., and Tambosso, T, (2005) 'Design and Optimization of a MEMS Electret-Based Capacitive Energy Scavenger'. *Journal of Microelectromechanical Systems*. vol 14, No 3, pp 429-435

[114] Xin, Z., Guangyi, S., Liang, R. and Guizhang, L. (2007) 'On MEMS Design Automation*'. Proceedings of the 26th Chinese Control Conference, 2007*, Zhangjiajie, Hunan, China. pp.774-778.

[115] Zha, X. and Du, H. (2003) 'Manufacturing process and material selection in concurrent collaborative design of MEMS devices'. *Journal of Micromechanics and Microengineering*, 13, pp. 509-522.

[116] Calis, M. and Desmulliez, M. (2006) 'Haptic Technologies for MEMS Design' *2006 Journal of Physics: Conference Series*, 34, pp.72-75.

[117] Gogoi, B., Yeun, R. and Mastrangelo, C, H, (1994) 'The Automatic Synthesis of Planar Fabrication Process Flows for Surface Micromachined Devices', *Proceedings IEEE Micro Electro Mechanical Systems Workshop*, Oiso, Japan, Jan. pp.153-157

[118] Shavezipur, M., Ponnambalam, K., Hashemi, S.M. and Khajepour, A, (2008) 'A probabilistic design optimisation for MEMS tunable capacitors'. *Journal of Microelectronics*. vol 39, 12, pp.1528-1533

[119] Cambell, M. L., (2001) 'An Automated Approach to Generating Novel MEMS Accelerometer Configurations' *Texas-Area MEMS Workshop,* Oral Presentation

[120] Cambell, M. L., (2000) *The A-Design Invention Machine: A Means of Automating and Investigating Conceptual Design*. Ph.D Thesis. Carnegie Mellon University, Carnegie Institute of Technology, United States of America

[121] Analog Devices, Inc., (1998), (online) "ADXL150/ADXL250: ±5 g to ±50 g, Low Noise, Low Power, Single/Dual Axis iMEMS® Accelerometers", *Datasheet*, http://www.analog.com/pdf/ADXL150_250_0.pdf, [accessed 11/2008].

[122] Kamalian, R.H., Agogino, A.M. and Takagi, H. (2004) 'The role of constraints and human interaction in evolving MEMS designs: microresonator case study', Proceedings of the ASME Design Automation Conference, Design Automation Track, Paper DETC2004-57462, CDROM, ISBN I710CD.

[123] Fan, Z., Wang, J. and Goodman, E.D. (2005) 'An evolutionary approach for robust layout synthesis of MEMS', Proceedings of 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp.1186–1191.

[124] Zhang, Y. (2006) 'MEMS design synthesis based on hybrid evolutionary computation', PhD thesis, Department of Civil and Environmental Engineering, University of California, Berkeley, USA.

[125] Cobb, C., Zhang, Y. and Agogino, A.M. (2006) 'An integrated MEMS design synthesis architecture using case-based reasoning and multi-objective genetic algorithms', Proceedings of 2006 SPIE Smart Materials, Nano- and Micro-Smart Systems, SPIE, Vol. 6414, No. 641419, ISBN: 9780819465221, Invited paper.

[126] Li, H. and Antonsson, E.K. (1998) 'Evolutionary techniques in MEMS synthesis', Proceedings DETC'98, 1998 ASME Design Engineering Technical Conferences, Atlanta, GA.

[127] Back, T., Hammel, U. and Schwefel, H. P. (1997). Evolutionary computation: Comments on the history and current state, IEEE Transactions on Evolutionary Computation 1(1): 3–17.

[128] Eiben, A. E., Hinterding, R. and Michalewicz, Z. (1999). Parameter control in evolutionary algorithms, IEEE Transactions on Evolutionary Computation3(2): 124–141. 2.2.1

[129] Fogel, D. B. (1997). The advantages of evolutionary computation, in D. Lundh, B. Olsson and A. Narayanan (eds), Bio-Computing and Emergent Computation 1997, World Scientific Press, Sk¨ove, Sweden, pp. 1–11

[130] Kamalian, R., and Agogino, A. M., (2005) “Improving Evolutionary Synthesis of MEMS through Fabrication and Testing Feedback” IEEE Systems, Man and Cybernetics (SMC) 2005, Waikoloa, HI, USA October

[131] Wang, K., and Nguyen, C. T-C. (1999) “High-Order Medium Frequency Micromechanical Electronic Filters” Journal of MicroElectroMechanical Systems. Vol 8. No 4. pp 534-556

[132] Zitzler, E., Laumanns, M., and Thiele, L., (2001) “SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization” *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, Athens, Greece: International Center for Numerical Methods in Engineering, pp 95-100

[133] Fan, Z., Seo, K, Hu, J., Goodmand, E. D., and Rosenberg, R. C. (2004) “A novel evolutionary engineering design approach for mixed-domain systems” Engineering Optimization, Vol 36. No 2, pp 127-147

[134] Moscato, P. (1993) “An introduction to population approaches for optimization and hierarchical objective functions: a discussion on the role of tabu search”. Annals of Operations Research, 41:85 121,

[135] Moscato, P. (1989) “On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms”. Technical Report Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA.

[136] Ma, L. and Antonsson, E.K. (2001) ‘Robust mask-layout synthesis for MEMS’, Technical Proceedings of the 2001 International Conference on Modeling and Simulation of Microsystems, Chapter 5: Optimization, pp.128–131.

[137] Zhou, N., Zhu, B., Agogino, A. M. and Pister, K. (2001) ‘Evolutionary Synthesis of MEMS (Microelectronic Mechanical Systems) Design’ , Proceedings of ANNIE 2001, IEEE Neural Networks Council and Smart Engineering Systems Laboratory, Nov. 4-7, 2001, Marriott Pavilion Hotel, St. Louis, Missouri, ASME Press, Vol. 11, pp.197-202.

[138] Fan, Z., Goodman, E.D., Wang, J., Rosenberg, R.C., Seo, K. and Hu, J. (2004) ‘Hierarchical evolutionary synthesis of MEMS’, Proceedings of the Congress on Evolutionary Computation (CEC 2004), Vol. 2, pp.2320–2327.

[139] Kamalian, R., Zhang, Y. and Agogino, A.M. (2005) ‘Microfabrication and characterizationof evolutionary MEMS resonators’, *In Proceedings of the Symposium of Micro- and Nano-Mechatronics for Information-based Society, IEEE Robotics and Automation Society*, November, ISBN 0-7803-9482-8, pp.109–114.

[140] Lohn, J. D., Kraus, W.F. and Hornby, G. S. (2007) ‘Automated Design of a MEMS Resonator’. Proceedings of the Congress on Evolutionary Computation. pp.3486-3491

[141] Hornby, G.S., Kraus, W.F. and Lohn, J.D. (2008) ‘Evolving MEMS resonator designs for fabrication’, Proceedings of the 8th international conference on Evolvable Systems: From Biology to Hardware, pp.213–224.

[142] Holland, J. H., (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor

[143] Goldberg, D. E., (1989), *Genetic Algorithms in Search, Optimization and Machine Learning,* Kluwer Academic Publishers, Boston, MA.

[144] Eiben, A. E., and Smith, J. E., (2003) Introduction to Evolutionary Computing, Springer, ISBN 3-540-40184-9

[145] modeFrontier (online) http://www.esteco.com/home/mode_frontier/mode_frontier.html [accessed 05/2009].

[146] Benkhelifa, E., Farnsworth, M., Tiwari, A. and Zhu, M., (2009) "An Integrated Framework for MEMS Design Optimisation using modeFrontier" EnginSoft International Conference 2009, CAE Technologies for Industry and ANSYS Italian Conference 2009.

[147] MathCAD (online) www.ptc.com/product/mathcad/ [accessed 11/2008].

[148] Zitzler, E., Knowles, J., and Thiele, L., (2008) "Quality Assessment of Pareto Set Approximations" LNCS Multiobjective Optimization. Springer-Verlag, Berlin Heidelberg, J. Branke *et al* (Eds). pp 373-404

[149] Fonesca, C. M., Paquete, L., and Lopez-Ibanez, M., (2006) "An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator" IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, pp 1157-1163.

[150] Hansen, M.P., Jaszkiewicz, A., (1998) "Evaluating the quality of approximations of the nondominated set", *Technical report*, Institute of Mathematical Modeling, Technical University of Denmark. IMM Technical Report IMM-REP-1998-7

[151] Knowles, J., Corne, D. (2002) "On Metrics for Comparing Non-Dominated Sets", In: Congress on Evolutionary Computation (CEC 2002), pp. 711–716. IEEE Press, Piscataway

[152] Ding, Y. (1988) "Multilevel Optimization of Frames with Beams Including Buckling Constraints" Computers & Structures. vol 32, No 2. pp249-261

[153] Walshaw, C. (2004) "Multilevel Refinement for Combinatorial Optimisation Problems" Annals of Operations Research, Vol 131, pp325-372

[154] Yammarino, F. J., and Dansereau, F. (2008) "Multi-Level nature of and multi-Level approaches to leadership" The Leadership Quarterly, Vol 19, pp135-141

[155] Yanhong, Z., Lei, Y., Hui, W., Feng, L., and Honghui, W. (2004) "Prediction of eukaryotic gene structures based on multilevel optimization" Chinese Science Bulletin, Vol 49, No 4, pp321-328

[156] Chan, T. F., Cong, J., Shinnerl, J. R., Sze, K., Xie, M., and Zhang, Y (2006) "Multiscale Optimization in VLSI Physical Design Automation" *Multiscale Optimization Methods and Applications*, Springer Publishers

[157] Nestor, J. A., and Thomas, D. E. (1982) "Defining and Implementing a Multilevel Design Representation With Simulation Applications" In 19th Design Automation Conference. pp 740-746

[158] Thomas, D. E., and Nestor, J. A. (1983) "Defining and Implementing a Multilevel Design Representation With Simulation Applications" IEEE Transactions on Computer-Aided Design, Vol CAD-2, No 3, pp135-145

[159] Hill, D., and vanCleemput, W. (1979) "SABLE: A Tool For Generating Structured, Multi-Level Simulations" In Proceedings of the 16th Design Automation Conference

[160] Bracken, J., and McGill, J. M. “Mathematical programs with optimization problems in the constraints” Operations Research. (1973) 21, pp37-44

[161] Chinchuluum, A., Pardalos, P. M., and Huang, H-X. (2009) “Multilevel (Hierarchical) Optimization: Complexity Issues, Optimality Conditions, Algorithms” Advances in Applied Mathematics and Global Optimization, eds Gao, D. Y., and Sherali, H. D. Advances in Mechanics and Mathematics, Vol 17, pp197-221

[162] Amouzegar, M. A., and Moshirvaziri, K. (1999) “Determining optimal pollution control policies: An application of bilevel programming” European Journal of Operational Research. 119, pp100-120

[163] Bard, J. F., Plummer, J., and Sourie, J. C. (2000) “A bilevel programming approach to determining tax credits for biofuel production” European Journal of Operational Research. 120, pp30-46

[164] Tiwari, A., Oduguwa, V., and Roy, Rajkumar. (2008) “Rolling System Design Using Evolutionary Sequential Process Optimization” IEEE Transactions on Evolutionary Computation, Vol 12, No 2, April, pp196-202

[165] El-Beltagy, M. A., and Keane, A. J. (1999) “A comparison of various optimization algorithms on a multilevel problem” Engineering Applications of Artificial Intelligence, pp 639-654

[166] Chong, S. Y., (2007) “Generalization and Diversity in Co-evolutionary Learning” Ph.D Thesis, University of Birmingham.

[167] Potter, M. A., and De Jong, K. A., (1994) “A cooperative coevolutionary approach to function optimization”. In Davidor. Y., Schwefel. H-P., Manner, R., editors. Proceedings of the Third International Conference on Parallel Problem Solving in Nature (PPSN-III), Jerusalem, Israel.

[168] Herrera, F., Lozano, M., and Moraga, C. (1997) “Hierarchical Distributed Genetic Algorithms” Technical Report #DECSAI-97-01-18

[169] Whitney, E. J., Sefrioui, M., Srinivas, K., and Periaux, J. (2002) “Advances in Hierarchical, Parallel Evolutionary Algorithms for Aerodynamic Shape Optimisation” JSME International Journal, Series B, Vol 45, No 1, pp 23-28

[170] Whitley, D., Rana, S., and Heckendorn, R. B., (1998) “The Island Model Genetic Algorithm: On Separability, Population Size and Convergence” Journal of Computing and Information Technology

[171] T., Starkweather, L. Darrell Whitley, and K. E., Mathias (1990) “Optimization Using Distributed Genetic Algorithms” In H.P. Schwefel and R. Manner, editors, *Parallel Problem Solving from Nature,* pp 176-185, Springer/Verlag

[172] Dunham, B., Fridshal, D., Fridshal, R., and North, J. H. (1963) “Design by Natural Selection” Synthese, Vol 15, pp254-259

[173] Goodman, E. D., Averill, R. C., Punch, W. F., and Eby, D. (1997) “Parallel Genetic Algorithms in the Optimization of Composite Structures” In Second World Conference (WSC2)

[174] Eby, D., Averill, R. C., Punch, W. F., and Goodman, E. D. (1998) “Evaluation of Injection Island GA Performance on Flywheel Design Optimization” In Proceedings of Third Conference on Adaptive Computing in Design and Manufacturing, I. Parmee, Ed. London: Springer Verlag, pp121-136

[175] Vekeria, H. D., and Parmee, I. (1997) “Co-operative Evolutonary Strategies for Single Component Design” In Proceedings of the Seventh International Conference on Genetic Algorithms” T. Back, Ed. San Fransico, CA: Morgan Kaufmann, pp 529-536

[176] Ding, Y., and Esping, B. J. D. (1991) “An Improved Multilevel Optimization Approach”, Computers & Structures. Vol 38, No. 5/6. pp 557-567

[177] Leary, S., Bhaskar, A., and Keane, A. (2001) “A Constraint Mapping Approach to the Structural Optimization of an Expensive Model using Surrogates” Optimization and Engineering, Vol 2, No 4, pp 385-398

[178] Maksimovic, S., and Zelijkovic, V. (2004) “Multilevel Optimization Approach Applied to Aircraft Structures” In The First International Conference on Computational Mechanics, pp 1-10

[179] Nair, P. B., and Keane, A. J. (1999) “Design Optimization of Space Structures With Nonperiodic Geometries for Vibration Suppression” AIAA

[180] Doorly, D. J., Piero, J., and Spooner, S. (1999) “Design Optimisation using distributed evolutionary methods” In 37th Aerospace Sciences Meeting and Exhibit AIAA-1999-111, Reno, NV

[181] Kampolis, I., Zymairs, A. S., Asouti, V. G., and Giannakoglou, K. C., (2007) “Multilevel optimization strategies based on metamodel-assisted evolutionary algorithms, for computationally expensive problems” In Congress on Evolutionary Computation, Singapore, New York, IEEE Press pp 4116-4123

[182] Kampolis, I.C., and Giannakoglou, K. C. (2010) “Synergetic use of different evaluation, parameterization and search tools within a multilevel optimization platform” Applied Soft Computing, doi: 10.1016

[183] Faisca, N. P., Saraiva, P. M., Rustem, B., and Pistikopoulos, E. N. (2009) “A Multi-parametric programming approach for multi-level hierarchical and decentralised optimisation problems”. Comput Manag Sci, 6, pp 377-397

[184] Grefenstette, J. J., and Fitzpatrick, J. M. (1985) “Genetic search with approximate function evaluations” In Proceeding of the First International Conference on Genetic Algorithms and Their Application pp 112-120

[185] Philipson, L. (1990) “Multilevel Design and Verification of Hardware / Software Systems” IEEE Journal of Solid-State Circuits, Vol 25, No 3, pp714-719

[186] Ellman, T,. Keane, J., Schwabacher, M., and Yao, K-T. (1997) “Multi-Level Modeling for Engineering Design Optimization” Artificial Intelligence for Engineering Design, Analysis, and Manufacturing

[187] Thiyagarajan, N., and Grandhi, R. V. (2005) “Multi-Level design process for 3-D perform shape optimization in metal forming” Journal of Materials Processing Technology. 170, pp 421-429

[188] Kampolis, I. C., and Giannakoglou K. C. (2009) “Distributed evolutionary algorithms with hierarchical evaluation” Engineering Optimization, Vol 41, No 11, pp1037-1049

[189] Lukas, D., and Chalmoviansky, P. (2007) “A sequential coupling of optimal topology and multilevel shape design applied to two-dimensional nonlinear magnetostatics” Comput Visual Sci, pp 135-144

[190] Lamberti, L. (2008) “An efficient simulated annealing algorithm for design optimization of truss structures” Computers and Structures, Vol 86, pp 1936-1953

[191] Li, Q. S., Lui, D. K., Zhang, N., Tam, C. M., and Yang, L. F. (2001) “Multi-level design model and genetic algorithm for structural control system optimization” Earthquake Engineering and Structural Dynamics, Vol 30, pp 927-942

[192] Desideri, J-A., and Janka, A., (2004) “Multilevel shape parameterization for aerodynamic optimization – application to drag and noise reduction of transonic / supersonic business jet” In European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004, Jyvaskyla, 24-28 July, 2004.

[193] Soper, A. J., Walshaw, C., and Cross, M. (2004) “A Combined Evolutionary Search and Multilevel Optimisation Approach to Graph-Partitioning” Journal of Global Optimization, Vol 29, pp225-241

[194] Korosec, P., and Silc, J. (2008) “The distributed multilevel ant-stigmergy algorithm used at the electric-motor design” Engineering Applications of Artificial Intelligence, Vol 21, pp941-951

[195] Liakopoulos, P. I. K., Kampolis, I. C., and Giannakoglou, K. C. (2008) “Grid enabled, hierarchical distributed metamodel-assisted evolutionary algorithms for aerodynamic shape optimization” Future Generation Computer Systems, Vol 24, pp701-708

[196] Hansen, L. U., and Horst, P. (2008) “Multilevel optimization in aircraft structural design evaluation” Computers and Structures, Vol 86, pp104-118

[197] Yui, K., Liu, Y., and Teo, K., “A hybrid descent method for global optimization” (2004) Journal of Global Optimization, 28. (2) pp 229-238

[198] Giannakoglou, K., Pappou, T., Giotis, A., Koubogiannis, D., (2000) “A parallel inverse-design algorithm in aeronautics based on genetic algorithms and the adjoint method” in ECCOMAS 2000, Barcelona, September

[199] Lee, Y., and Lee, M., (2002) “A Shape-based block layout approach to facility layout problems using hybrid genetic algorithm” Comput. Ind. Engrg, 42 (2) pp 237-248

[200] Svanberg, K. (1987) “Method of moving asymptotes – a new method for structural optimization” Intl. J. Numer. Meth. Engng. 24, pp 359-373

[201] Li, G., Zhou, R-G., Duan, L., and Chen, W-F. (1999) “Multiobjective and multilevel optimization for steel frames” Engineering Structures, Vol 21, pp519-529

[202] Conceicao Antonio, C. A. (2002) “A multilevel genetic algorithm for optimization of geometrically nonlinear stiffened composite structures” Struct Multidisc Optim, Vol 24, pp372-386

[203] Litovski, V., and Andrejevic, M., (2005) “Behavioural Modelling, Simulation, Test and Diagnosis of MEMS using ANNs” In, *International Symposium on Circuits and Systems, Kobe, Japan, 23 - 26 May 2005.* IEEE.

[204] Schneider, P., Schneider, A., Bastian, J., Reitz, S., and Schwarz, P., (2002) “MOSCITO – A Program System for MEMS Optimization” DTIP, Cannes, May.

[205] Karakasis, M., and Giannakogolou, K., (2005) “On the use of metamodel-assisted, multi-objective evolutionary algorithms” Engineering Optimisation. 38(8) pp 941-957

[206] Ong, Y. S., Nair, P. B., Keane, A. J., (2003) ”Evolutionary optimization of computationally expensive problems via surrogate modelling” Journal AIAA. 41(4) pp 687-696

[207] Haykin, S., (1999) “Neural Networks: A Comprehensive Foundation, second ed., Prentice Hall, New Jersey, USA.

[208] Giannakoglou, K., Papadimitrou, A., and Kampolis, I. (2006) “Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels” Computer Methods in Applied Mechanics and Engineering, 195, (44-47), pp 6312-6329,

[209] Myers, R. H., and Montgomery, D. C., (2002) “Response Surface Methodology: Process and Product Optimization Using Designed Experiments, Probability and Statistics” second edition, John Wiley & Sons; New York, USA.

[210] Simpson, T., Mauery, T., Korte, J., Mistree, F., (1998) “Comparison of response surface and kriging models for multidisciplinary design optimization” AIAA paper 98-4755

[211] Balling, R., and Rawlings, M. W. (2000) “Collaborative optimization with disciplinary conceptual design” Structural Multidisciplinary Optimisation, Vol 20, pp 232-241

[212] Tribes, C., Dube, J-F., and Trepanier, J-Y. (2005) “Decompostion of multidisciplinary optimization problems: formulations and applications to a simplified wing design” Engineering Optimization Vol 37, No 8, pp775-706

[213] de Wit, A. J., and van Keulen, E. (2010) “Overview of Methods for Multi-Level and/or Multi-Disciplinary Optimization” In 6th AIAA Multidisciplinary Design Optimization Specialist Conference, AIAA Paper Number 2010-2914

[214] Cramer, E. J., Dennis, J. E., Frank, P. D., Lewis, R. M., Shubin, G. R., (1994) “Problem Formulation for multidisciplinary optimization” SIAM J Optim, Vol 4, pp 754-776

[215] Kroo, I., and Manning, V. (2000) “Collaborative Optimization: Status and Directions” In 8th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Paper Number AIAA 2000-4721

[216] Schoeffler, J. D. (1971) “Optimization methods for large-scale systems … with applications” chapter 1. Static multilevel systems. McGRAW-HILL BOOK COMPANY

[217] Sobieszczanski-Sobieski, J. (1990) “Sensitivity analysis and multidisciplinary optimization for aircraft design: Recent advances and results” Journal of Aircraft, Vol 27, pp993-1001

[218] Wagner, T.C (1993) “A general decomposition methodology for optimal design” PhD thesis, The University of Michigan

[219] Balling, R. J., and Sobieszczanski-Sobieski, J. (1996) “Optimization of coupled systems: A critical overview of approaches” American Institute of Aeronautics and Astronautics Journal, Vol 34.

[220] Sobieszczanski-Sobieski, J., and Haftka, R. T. (1997) “Multidisciplinary aerospace design optimization: survey of recent developments” Structural Optimization, Vol 14, pp1-23

[221] Kodiyalam, S., and Sobieszczanski-Sobieski, J. (2001) “Multidisciplinary design optimization – some formal methods, framework requirements, and application to vehicle design” International Journal of Vehicle Design, Vol 23, pp 3-22

[222] Sobieszczanski-Sobieski, J., James, B. B., and Dovi, A. R. (1985) “Structural optimization by multi-level decomposition” American Institute of Aeronautics and Astronautics Journal, Vol 23, pp 124-142

[223] Braun, R., Gage, P., Kroo, I., and Sobieski, I. "Implementation and performance issues in collaborative optimisation" In Proceedings of the 6th AIAA/USAF/NASA/ISSMO Symp. On Multidisciplinary Analysis and Optimisation (1996), AIAA Paper Number 96-4017

[224] Sobieszczanski-Sobieski, J. (1988) "Optimization by decomposition: a step from hierarchic to non-hierarchic systems" Recent Advances in Multidisciplinary Analysis and Optimization, NASA CP-3031

[225] Sobieszczanski-Sobieski, J., Argte, J. S., and Sandusky, R. R., Jr. (1998) "Bi-Level integrated system synthesis (BLISS) Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symp. On Multidisciplinary Analysis and Optimization AIAA Paper Number 98-4916

[226] Kim, H. M., Nestor, F. M., Panos, P. Y., and Tao, J. (2003) "Target cascading in optimal system design" Journal of Mechanical Design, Vol 125, pp474-480

[227] Haftka, R. T., and Watson, L. T. (2005) "Multidisciplinary Design Optimisation with Quasiseparable Subsystems" Optimization and Engineering, Vol 6, pp9-20

[228] DeMiguel, A. V., and Murray, W. (2006) "A local convergence analysis of bilevel decomposition algorithms" Optimization and Engineering, Vol 7, pp99-133

[229] Tosserams, S., Etman, L. F. P., Rooda, J. E., (2008) "Augmented Lagrangian coordination for distributed optimal design in MDO" International Journal of Numerical Methods in Engineering, Vol 73, pp1885-1910

[230] de Wit, A. J., Lipka, A., Ramm, E., and van Keulen, E. (2006) "Multilevel Optimization of Material and Structural layout" In 3rd European Conference on Computational Mechanics, Solids, Structures and Coupled Problems in Engineering

[231] Sobieszczanski-Sobieski, J. (1982) "A linear decomposition method for large optimization problems. Blueprint for development" Technical Report, NASA-TM-83248, NASA Feb

[232] Sobieszczanski-Sobieski, J., James, B. B., and Riley, M. F., (1987) "Structural sizing by generalized, multilevel optimization" American Institute of Aeronautics and Astronautics Journal, Vol 25, pp139-145

[233] Wrenn, G. A., and Dovi, A. R. (1988) "Multilevel decomposition approach to the preliminary sizing of a transport aircraft wing" Journal of Aircraft Vol 25, pp 223-230

[234] Chattopadhyay, A., McCarhty, T. R., and Pagaldipti, N. (1995) "Multilevel decomposition procedure for efficient design optimization of helicopter rotor blades" American Institute of Aeronautics and Astronautics Journal, Vol 33, pp223-230

[235] Li, Q. S., Liu, D. K., Fang, J. Q., and Tam, C. M. (2000) "Multi-level optimal design of buildings with active control under winds using genetic algorithms" Journal of Wind Engineering and Industrial Aerodynamics Vol 86, pp 65-86

[236] Chen, T.-Y., and Yang, C.-M (2005) "Multidisciplinary design optimization of mechanisms" Advances in Engineering Software Vol 86, pp 65-86

[237] Sobieszczanski-Sobieski, J. (1993) "Two alternative ways for solving the coordination problem in multilevel optimization" Structural Optimization, Vol 6, pp 205-215

[238] Balling, R. J., and Sobieszczanski-Sobieski, J. (1995) "An algorithm for solving the system-level problem in multilevel optimization" Structural Optimization, Vol 9, pp 168-177

[239] Renuad, J. E., and Gabriele, G. A. (1993) "Improved Coordination in Nonhierarchic System Optimization" American Institute of Aeronautics and Astronautics Journal, Vol 31, pp 2367-2373

[240] Renuad, J. E., and Gabriele, G. A. (1994) "Approximation in Nonhierarchic System Optimization" American Institute of Aeronautics and Astronautics Journal, Vol 32, pp198-205

[241] Stelmack, M. A., Batill, S. M., Beck, B. C., and Flask, D. J. (1998) "Application of the concurrent subspace design framework to aircraft brake component design optimization" AIAA/ASME/ASCE/AHS/ASC 38th Structures, Structural Dynamics and Materials Conference, Long Beach California, AIAA Paper Number 98-2033

[242] Braun, R. D., Moor, A. D., and Kroo, I. M (1997) "Collaborative approach to launch vehicle design" Journal of Spacecraft and Rockets, Vol 34, pp478-486

[243] Budianto, I. A., and Olds, J. R. (2004) "Design and deployment of a satellite constellation using collaborative optimization" Journal of Spacecraft and Rockets, Vol 41, pp956-963

[244] Sobieszczanski-Sobieski, J., and Kodiyalam, S. (2001) "Bliss/s: a new method for two-level structural optimization" Structural and Multidisciplinary Optimization, Vol 21, pp-1-13

[245] Sobieszczanski-Sobieski, J., Altus, T. D., Phillips, M., and Sandusky, R. (2003) "Bilevel integrated system synthesis for concurrent and distributed processing" American Institute of Aeronautics and Astronautics Journal, Vol, 41, pp 1996-2003

[246] de Beats, P. W. G., Mavris, D. N., and Sobieszczanski-Sobieski, J., (2004) "Aeroelastic design by combining conventional practice with bi-level integrated system synthesis (bliss)" In 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, number 2004-4431 in AIAA

[247] Kokkolaras, M., Fellini, R., Kim, H. M., Michelena, N. F., and Papalambros, P. Y. (2002) "Extension of the target cascading formulation to the design of product families" Structural and Multidisciplinary Optimization, Vol 24, pp 293-301

[248] Allison, J., Kokkolaras, M., Zawislak, M., and Papalambros, P. Y. (2005) "On the use of analytical target cascading and collaborative optimization for complex system design" In 6th World Congress on Structural and Multidisciplinary Optimization

[249] Choudhary, R., Malkawi, A., and Papalambros, P. Y. (2005) "Analytical Target cascading in simulation-based building design" Automation in Construction, Vol 14, pp551-568

[250] Lui, B., Haftka, R. T., and Akgun, M. A. (2000) "Two-level composite wing structural optimization using response surfaces" Structural and Multidisciplinary Optimization, Vol 20, pp87-96

[251] Lui, B., Haftka, R. T., and Watson, L. T. (2004) "Global-local structural optimization using response surfaces of local optimization margins" Structural and Multidisciplinary Optimization, Vol 27, pp352-359

[252] Tapetta, R. V., and Renaud, J. E. (1997) "Multiobjective Collaborative Optimization" Journal of Mechanical Design, ASME Transactions, Vol 119, pp 403-411

[253] McAllister, C. D., Simpson, T. W., Hacker, K., Lewis, K., and Messac, A. (2005) "Integrating linear physical programming within collaborative optimization for multiobjective multidisciplinary design optimization" Structural Multidisciplinary Optimization, Vol 29, pp178-189

[254] Kurapati, A., and Azarm, S. (2000) "Immune Network Simulation with MultiObjective Genetic Algorithms for Multidisciplinary Optimization" Engineering Optimization, Vol 33, pp245-260

[255] Gunawan, S., Azarm, S., Wu, J., and Boyars, A. (2003) "Quality-Assisted Multi-Objective Multidisciplinary Genetic Algorithms" AIAA Journal, Vol 41, No 9. Pp 1752-1762

[256] Giassi, A., Bennis, F., and Maisonneuve, J-J. (2004) "Multidisciplinary design optimisation and robust design approaches applied to concurrent design" Structural Multidisciplinary Design, Vol 28, pp 356-371

[257] Aute, V., and Azarm, S. (2006) "A Genetic Algorithms Based Approach for Multidisciplinary Multiobjective Collaborative Optimization" In 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Paper Number AIAA 2006-6953

[258] Rabeau, S., Depince, P., and Bennis, F. (2007) "Collaborative optimization of complex systems: a multidisciplinary approach" International Journal Interactive Design Manufacturing, Vol 1, pp209-218

[259] Huang, H-Y., and Wang, D-Y. (2009) "Static and dynamic collaborative optimization of ship hull structure" Journal of Marine Science Applications, Vol 8, pp77-82

[260] Zadeh, P, M., Roshanian, J., and Farmani, F, R. (2010) "Particle Swarm Optimization for Multiobjective Collaborative Multidisciplinary Design Optimization" In 51st AIAA/ASMEA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA Paper Number 2010-3081

[261] Gunawan, S., Farhang-Mehr, A., and Azarm, S., (2004) "On Maximizing Solution Diversity in a Multiobjective Multidisciplinary Genetic Algorithm for Design Optimization" Mechanics Based Design of Structures and Machines, Vol 32, No. 4, pp. 491-514

[262] Takama, N., and Loucks, D. P. (1981) "Multi-Level Optimization for multi-objective problems" Appl. Math. Modeling. Vol 5, pp173-178

[263] Lee, J., and Hajela, P. (1997) "GA's in decomposition based design-subsystem interactions through immune network simulation" Struct. Optim, Vol 14, pp248-255

[264] Conceicao Antonio, C. A. (1999) "Optimization of geometrically nonlinear composite structures based on load-displacement control" Composite Struct, Vol 46, pp345-356

[265] Suh, N. P., (2001) "Axiomatic Design: Advances and Applications", Oxford University Press, 2001, ISBN 0-19-513466-4

[266] Agarwal, M. and Cagan, J., (1999) "Systematic form and function design of MEMS resonators using shape grammars." ICED'99, Technische Universität München.

[267] Agarwal, M., Cagan, J., and Stiny, G., (2000) "A micro language: generating MEMS resonators using a coupled form-function shape grammar," Environment and Planning B: Planning and Design, vol. 27, pp. 615-626.

[268] Microsoft C# (Online) http://msdn.microsoft.com/en-us/vstudio/hh388566.aspx [accessed 11/2008].

[269] Microsoft .NET (Online) http://www.microsoft.com/net [accessed 11/2008].

[270] J.J. Durillo, A.J. Nebro (2011) "jMetal: a Java Framework for Multi-Objective Optimization" Advances in Engineering Software 42, pp 760-771

[271] Golinski, J., (1970) "Optimal synthesis problems solved by means of nonlinear programming and random methods" Journal of Mechanisms, Vol 5, pp 287-309

[272] Nguyen, C, T-C., Katchi, L. P. B., and Rebeiz, G. M., (1998) "Micromachined Devices for Wireless Communications" Proceedings IEEE, vol 86, no 8, pp. 1756-1768, Aug

[273] Mestrom, R. M. C., (2007) "Microelectromechanical Oscillators – a literature survey" DCT Internal report, Eindhoven University of Technology,

[274] Bannon. F. D., Clark, J. R., and Nguyen, C. T-C., (2000) "High-Q HF Microelectromechanical Filters" IEEE Journal of Solid-State Circuits, Vol 35. No 4. pp 512-526

[275] Jing, Q., Mukherjee, T., and Fedder, G., (1999) "A Design Methodology for Micromechanical Bandpass Filters" In Proceedings of 1999 IEEE/ACM/VIUF International Workshop on Behavioral Modeling and Simulation (BMAS '99), Oct 4-6, Orlando, Fl, USA

[276] Sykes, R. A., Smith, W. L., and Spencer, W. J., (1967) "Monolithic crystal filters" in IEEE International Conv. Rec. pt. II, March. pp 78-93

[277] Rennick, R. C., (1973) "An equivalent circuit approach to the design and analysis monolithic crystal filters" IEEE Transactions. Sonic. Ultrason., vol SU-20, pp 347-354

[278] Campbell, C. K., (1998) "Surface Acoustic Wave Devices for Mobile Wireless Communications" New York, NY:Academic

[279] Krishnaswamy, S. V., Rosenbaum, J., Horwitz, S., Yale, C., and Moore, R. A., (1991) "Compact FBAR Filters offer low-loss performance" Microwaves and RF, pp. 127-136, Sept

[280] Dec, A., and Suyama, K., (2000) "Microwave MEMS-based Voltage-controlled Oscillators" IEEE Transactions on Microwave Theory and Techniques, Vol, 48. No 11. Part I. pp. 1943-1949, Nov

[281] Yao, J. J., and Chang, M. F., (1994) "A Surface Micromachined Miniature Switch for Telecommunications Applications with Signal Frequencies from DC up to 4 GHz" 8th International Conference on Solid State Sensors and Actuators and Eurosensors IX. pp. 384-387

[282] Kwon, Y., Cheon, C., Kim, N., Kim, C., Song, I., Song, C., (1999) "A Ka-band MMIC Oscillator Stabalized with a Micromachined Cavity" IEEE Microwave and Guided Wave Letters, Vol 9, No 9. pp 360-362 Sept

[283] Brown, A. R., and Rebiez, G. M., (1999) "A High Performance Integrated Band Diplexer" IEEE Transactions on Microwave Theory and Techniques, Vol 47, No 8. pp 1477-1481, Aug.

[284] De Los Santos, H. J., and Richards, R. J., (2001) "MEMS for RF/Microwave Wireless Applications: The Next Wave – Part II" Microwave Journal. July

[285] Brank, J., Yao, J., Eberly, M., Malczewski, Varian, K., and Goldsmith, C., (2001) "RF MEMS-Based Tunable Filters" John Wiley & Sons, Inc

[286] Nguyen, C. T-C (1999) "Micromechancial filters for miniaturized low-power communications" Proceedings of SPIE: Smart Structures and Materials (Smart Electronics and MEMS), Newport Beach, California, March.

[287] Lin, L., Nguyen, C. T-C., Howe, R. T., and Pisano, A. P., (1992) "Micro Electromechanical Filters for Signal Processing" Micro Electro Mechanical Systems

[288] Nguyen, C. T-C., "Micromechanical Resonators for Oscillators and Filters" Proceedings of the 1995 IEEE International Ultrasonics Symposium, Seattle, WA, pp 489-499, Nov-7-10. (1995)

[289] Tilmans, H. A. C., (1996) "Equivalent circuit representation of electromechanical transducers: I. Lumped-parameter systems" Journal Micromechanical Micro engineering. Vol 6, pp 157-176

[290] Brotz, J., "Damping in CMOS-MEMS Resonators" MSc Thesis, Carnegie Mellon University (2004)

[291] David H. Eberly, (2000) "3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics", Morgan Kaufmann Publishers, San Francisco, CA

[292] Haemer J. M., Sitaraman, S. K., Fork, D. K., Chong, F. C., Mok, S., Smith, D. L., Swiatowiec, F., "Flexible micro-spring interconnects for high performance probing" (2000) Proceedings of the 50th Electronic Components and Technology Conference, May 21-24, pp 1157-1163

[293] Ozevin, D., Pessiki, S., Jain, A., Greve, D.W., and Oppenheim, A. (2003) J., SPIE Smart Structures Conference SN09: Smart and NDE for Civil Infrastructure, San Diego http://www.ece.cmu.edu/~dwg/research/spieaefinal.pdf.

[294] Park, Kyu-Yeon., Lee, Chong-Won, Oh, Yong-Su., and Cho, Young-Ho., (1998) "Laterally oscillated and force balanced micro vibratory rate gyroscope supported by fish hook shape springs" Sensors and Actuators A (Physical) A64 (1) pp 69-76

[295] Weinstein, D., Bhave, S. A., Tada, M., Mitarai, S., and Ikeda, K., (2007) "Mechanical Coupling of 2D Resonator Arrays for MEMS Filter Applications" IEEE pp 1362-1365

[296] Saad, N. H., Al-Dadah, R. K., Anthony, C. J., and Ward, M. C.L., (2009) "Analysis of MEMS mechanical spring for coupling multimodal micro resonators sensor" Microelectronic Engineering. 86 pp 1190-1193

[297] Miyamoto, K., Jomori, T., Sugano, K., Tabata, O., and Tsuchiya, T., (2008) "Mechanical calibration of MEMS springs with sub-micro-Newton force resolution" Sensors and Actuators A. Vol 143, pp 136-142

[298] Wang, F., Keskin, G., Phelps, A., Rotner, J., Li, X., Fedder, G.K., Mukherjee, T., and Pileggi, L.T., (2012) "Statistical Design and Optimization for Adaptive Post-silicon Tuning of MEMS Filters" Proceedings of the 49th Annual Design Automation Conference, DAC'12, pp 176-181